

DEVELOPMENT OF A FINITE ELEMENT ANALYSIS
PROGRAM FOR STRUCTURAL DYNAMICS
APPLICATIONS

By

MUHAMMET ALI SAGLAR

Bachelor of Civil Engineering

Istanbul Technical University

Istanbul, Turkey

2007

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2009

DEVELOPMENT OF A FINITE ELEMENT ANALYSIS
PROGRAM FOR STRUCTURAL DYNAMICS
APPLICATIONS

Thesis Approved:

Dr. Jonathan S. Goode

Thesis Adviser

Dr. Robert N. Emerson

Dr. Tyler Ley

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

I thank God for His grace, His blessings and guiding me to the right path.

I would like to express profound gratitude to my advisor, Dr. Jonathan S. Goode, for his precious support, encouragement and useful suggestions throughout this research work. His moral support and continuous guidance enabled me to complete my work successfully.

I am also highly thankful to Dr. Robert Emerson and Dr. Tyler Ley for their encouragements guidance rendered to me throughout my research. Their deep experience helped me a lot to be a structural engineer.

I would also like to thank to Rameez Iqbal, M.S student, for his support during this research. I am really grateful to him for his friendship and understanding.

My deepest appreciation goes to my parents, Remzi Saglar and Asiye Saglar for their invaluable supports during every level of my life. I am grateful also to my three brothers their unflinching courage and conviction.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
1.1 Background and Problem Definition	1
1.2 Objectives	2
1.3 Overview	3
II. LITERATURE REVIEW	6
2.1 Finite Element Method	6
2.2 Structural Dynamics	8
2.3 MATLAB Usage in Structural Dynamics	10
III. FINITE ELEMENT THEORETICAL DERIVATIONS	12
3.1 Introduction	12
3.2 Interpolation and Shape Functions	13
3.3 Formulation of Element Stiffness Matrices	16
3.4 Element Boundary and Loading Conditions	17
3.5 Solution Techniques	19
3.6 Structural Dynamics	21
3.6.1 Equation of Motion	21
3.6.2 Formulation of Element Mass Matrices	22
3.6.3 Dynamic Loading Conditions	23
3.6.4 Numerical Integration	26
3.6.5 Newmark-Beta Solution Technique	27
IV. CASE STUDIES	29
4.1 Introduction	29
4.2 Harmonic Loading	31
4.3 Impulse Loading	36
4.4 Step Loading	42
4.5 Multiple Dynamic Loads	47

Chapter	Page
V. CONCLUSION	54
5.1 Summary and Conclusions	54
5.2 Recommendations	55
REFERENCES	57
APPENDIX A – MATLAB FEA PROGRAM.....	59
APPENDIX B – CONFIRMATION CALCULATIONS.....	68

LIST OF TABLES

Table	Page
4-1 Case Studies	30

LIST OF FIGURES

Figure	Page
2-1 Crack Distributions at the Ultimate Load Step (Ozcan, 2008)	8
3-1 Q4 Finite Element Nodal Degrees of Freedom.....	15
3-2 Harmonic Loading	24
3-3 Impulse Loading	25
3-4 Step Loading	25
4-1 Case Study 1 – Cantilever Beam	31
4-2 Case Study 1 – Harmonic Loading	32
4-3 Case Study 1 – Finite Element Mesh.....	33
4-4 Case Study 1 – Displacements.....	33
4-5 Case Study 2 – Simply-Supported Beam.....	37
4-6 Case Study 2 – Impulse Loading	37
4-7 Case Study 2 – Finite Element Mesh.....	38
4-8 Case Study 2 – Displacements.....	39
4-9 Case Study 3 – Continuous Beam.....	42
4-10 Case Study 3 – Step Loading.....	43
4-11 Case Study 3 – Finite Element Mesh.....	44
4-12 Case Study 3 – Displacements.....	44
4-13 Case Study 4 – Continuous Beam.....	48
4-14 Case Study 4 – Multiple Loading	49
4-15 Case Study 4 – Finite Element Mesh.....	50
4-16 Case Study 4 – Displacements.....	50

CHAPTER I

INTRODUCTION

1.1 Background and Problem Definition

Finite element analysis (FEA) is a computational technique used to find approximate solutions of field problems that are described by differential equations or by an integral expression. The finite element solution technique helps in solving complex field problems by numerical approximations of the differential or integral expression.

Finite element analysis has significant advantages when compared to the other numerical analysis methodologies. It is very powerful and applicable in many engineering problems such as displacements of a structural systems, stress-strain analysis, heat transfer and magnetic fields. In addition, for individual elements different material properties can be incorporated. One of the most important advantages of finite element analysis is that there are no limitations concerning the geometry or boundary conditions. Different types of geometry and boundary conditions can be accommodated easily. In addition to these properties, it is easy to modify the problem and increase the accuracy of the results while usually only at the expense of computational efficiency.

At present many commercial software packages use the finite element analysis because of the advantages mentioned above. ANSYS, ALGOR, ABAQUS, COSMOS/M and SAP2000 are some of the well-known commercial software packages that are based on finite element analysis. That is to say that the finite element analysis has wide range of usage and is conveniently available for engineers and researchers.

For research purposes, finite element analysis is very useful to allow for examination of “what if” design scenarios. As mentioned earlier, it has almost no limitation for engineering problems in terms of geometry and boundary conditions. Because of these properties it is very powerful for researchers.

In this study, four types of dynamic loads are applied separately to different beams. The finite element analysis is used to define the beam and determine the problem solution for both loading cases. Furthermore, Q4 finite elements are considered during the analysis and these elements are combined to form a mesh. During the dynamic analysis calculation procedure a numerical procedure, Newmark’s method, is used to numerically integrate the equations of motion with respect to time.

1.2 Objectives

The primary objective of this research is the development of a finite element analysis computer program written using MATLAB. This program will be used to determine structural displacements or response of structural systems to both static and dynamic

loading conditions. As a method of illustrating the use of this program, several case studies are presented that find the displacements and deformed shape of a beam under the dynamic loading with change of time by using finite element method. The purpose of this study is development of a coupled finite element analysis/structural dynamics computer program. This study does not focus on accuracy of the results. In other words, the best approximation to the exact results is not the main intention. As previously mentioned, the computer codes are written in MATLAB which is a computationally powerful programming language for research purposes.

Explained in its most basic form, this computer code has the capability to calculate approximate displacements which can be calculated using design software packages. Besides these capabilities, the development of this program gives the author the understanding of what is going on behind the software packages. This program has the capacity to incorporate different scenarios for dynamic loading. In other words, any time-dependent loading scenario to calculate the dynamic load can be applied to the beam to obtain displacements and the deformed shape of the structure.

1.3 Overview

The next several chapters present the methodology behind this study. A summary of each chapter is provided to give a brief overview of the remaining sections of this study:

- **Chapter 2 – Literature Review:**

Current and recent studies are provided from a broad viewpoint including a brief discussion about the usage of the finite element method and structural dynamics. The literature review gives several examples that use the finite element method, structural dynamics and structural dynamics combined with the finite element method.

- **Chapter 3 – Finite Element Theoretical Derivations:**

Detailed theoretical development of the finite element analysis procedure and structural dynamics is presented and discussed. This information is directly used to develop the MATLAB code for the FEA program. Theoretical equations are derived and explained in this chapter. Topics that are covered include discussions about generating element matrices, boundary and loading conditions, finite element solution techniques, and structural dynamics solution technique.

- **Chapter 4 – Case Studies:**

The capabilities of the MATLAB FEA program developed as a consequence of this study are demonstrated. Four case studies considering a dynamic loading are demonstrated on different structural beam configurations. Hand calculations (included in Appendix B of this study) are provided for confirmation of the FEA program.

- **Chapter 5 – Summary, Conclusion and Recommendations:**

A summary of the results and conclusions of the study are provided. This includes a brief discussion and capability of this study. In addition, recommendations about the study and its expandability are provided.

CHAPTER II

LITERATURE REVIEW

2.1 Finite Element Method

The finite element method is an approximation method that can be used to calculate stresses, movements of loads and forces, displacements, heat transfer and other basic physical behaviors while using very large matrix arrays and mesh diagrams. In recent years, the finite element method has been used to obtain the solutions to a variety of engineering problems. A few applications of finite element modeling that have been used for engineering research are subsequently illustrated.

Li et al. (2001) presented a quadratic finite element considering the principle that the local displacement fields of the elements are compatible with the global displacement field of the corresponding systems by using generalized degrees of freedom (GDOF) and a quadratic finite strip with GDOF. They developed a global displacement field based on quadratic B-spline functions and created local displacement field strips for the elements. Therefore, they formed models for the finite elements and finite strips corresponding to the GDOF by rearranging the local displacement fields of elements and strips to be compatible with their corresponding global displacement fields. In their study, several

numerical examples were provided for authenticity, simplicity and versatility of the element and strip in the analysis of thin-walled structures. The authors determined that the finite element and finite strip with GDOF can handle inconveniences in the analysis of beams, plates and other structures when the thickness changes. In addition to handling such problems, the authors also determined that the finite element and finite strip gave similar results with fewer degrees of freedom when compared to traditional finite element solutions.

Özcan et al. (2008) presented three steel fiber-added reinforced concrete models which were analyzed experimentally and numerically with a finite element analysis. They created a finite element model using ANSYS. The authors considered eight-node solid brick elements in their finite element modeling. Experimental data and finite element analysis results were compared. Figure 2-1 is given as an illustration for comparison of experimental research and finite element modeling. As a result of their study, the experimental data and finite element modeling gave very close results in terms of deflections and stresses at the center line along with initial and progressive cracking, failure mechanism and load, deflections and stresses at the zero deflection point and decompression.

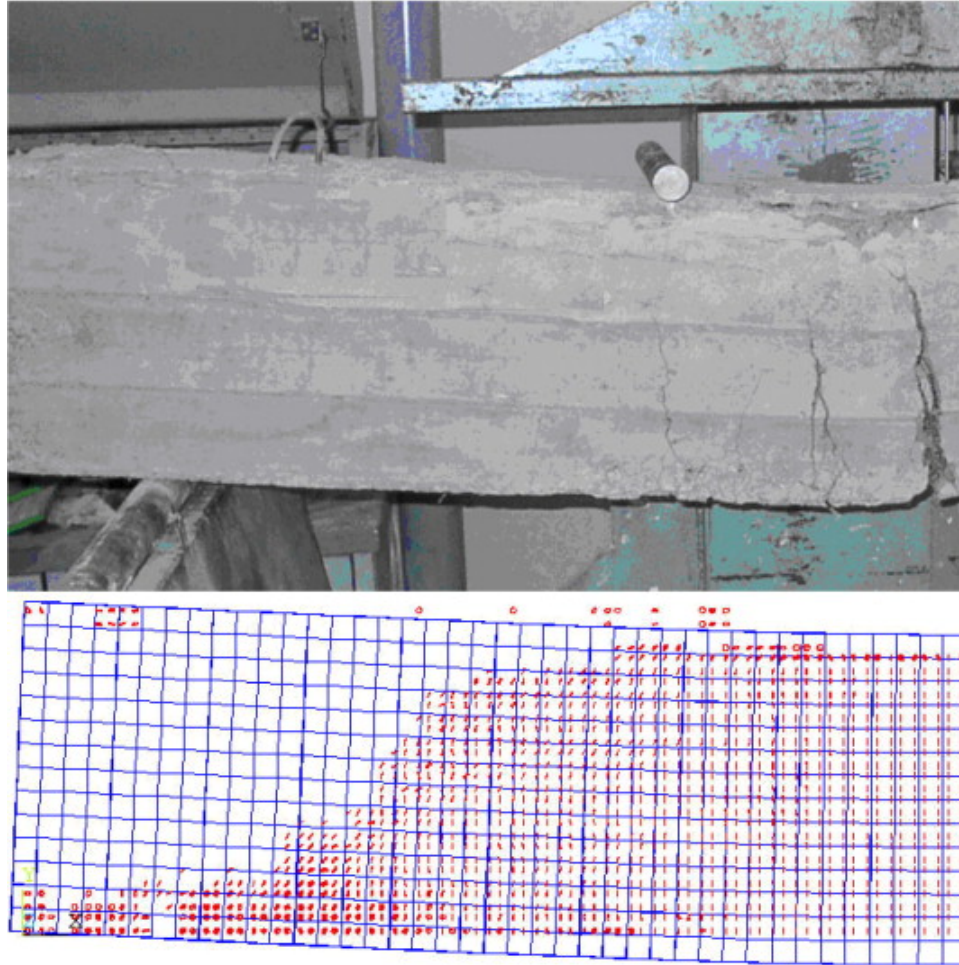


Figure 2-1: Crack Distributions at the Ultimate Load Step (Ozcan, 2008)

2.2 Structural Dynamics

Incorporation of the dynamic response of structural systems into a finite element model is a natural extension of the numerical time-stepping methods for dynamic analysis. Recently, many studies have considered this kind of “coupling” of a FEA program and structural dynamics models. A review of some of these studies is provided herein.

Du et al. (1992) considered a three-dimensional elastic beam with an arbitrary and large moving base with six degrees of freedom by creating a general finite element structural dynamics model. In their study, six degrees of freedom of the beam base may include either a particular arbitrary motion of the base or the coupling of the beam with other structures. The beam can be pre-twisted and has a mass center offset from the elasticity center. In the study, the authors derived the equations of motion using the principle of virtual work in the finite element analysis. To simplify the analysis, the authors combined the beam inertia at the end nodes of each element. At the conclusion of the study, it was determined that the model provided flexibility because of the combination of the finite element method and arbitrary base motion variables that were used in multi-body dynamics and a fundamental element to solve the dynamic problems of rotating beam-like structures.

Cerioni et al. (1995) presented a finite element model that was capable of analyzing the dynamic nonlinear behavior of unreinforced masonry panels in a biaxial stress field by using a non-conforming quadrilateral element. The study derived the equilibrium equations including the inertial and damping actions by a direct step-by-step integration procedure in the time domain known as Newmark's method. The results were compared with experimental results in terms of displacement, velocity, acceleration, strain and stress. The comparison of the results indicated good performance not only for non-conforming quadrilateral elements but also time-dependent and nonlinear analyses. In addition to these results, the model provided a very simple, computationally economical and convenient analysis of complex structural problems.

Mermertas et al. (1997) introduced a curved bridge deck to examine the dynamic interaction between a vehicle that is assumed to have four-degrees of freedom and a curved beam. The finite element method was used to model the curved beam assumed to be simply supported neglecting any damping of the structure. The authors also applied a multi-predictor-corrector procedure in conjunction with the Newmark method in the solution of the resulting equations of motion. The authors determined the mid-span deflection of the bridge for different vehicle speeds and varying radii of curvature for the curved deck.

2.3 MATLAB Usage in Structural Dynamics

The usage of computer programs to use the finite element method and numerical time-stepping methods for the structural dynamics has become increasingly important in recent years. Advances in computer technology allow even simple computer systems to model very complex systems with ease and efficiency. A popular program in research studies is MATLAB. This program can be utilized very easily to accomplish both a finite element analysis and structural dynamics analysis due to its friendly matrix manipulations. Several recent research studies have considered MATLAB and they are further outlined here.

Kiral et al. (2008) presented a fixed-fixed laminated composite beam that was subjected to a concentrated force traveling at a constant velocity to determine the dynamic behavior

of a beam. The authors used classical lamination theory in order to create a three-dimensional finite element model. In addition, the Newmark integration method to compute the dynamic response was implemented in MATLAB. The dynamic magnification, defined as the ratio between the dynamic and static displacements, was determined from their study. As a result of the study, it was determined that the load velocity and ply orientation may affect the dynamic response significantly. It was also determined that if the total traveling time is equal to the first natural period of the beam, the maximum deflection occurs at the mid-point of the beam.

Bozdogan et al. (2009) demonstrated an approximate method using a continuum approach and transfer matrix method for static and dynamic analyses of multi-bay coupled shear walls. The authors assumed the structure system as a sandwich beam and thus wrote the differential equation whose solution gave the shape functions for each story of the sandwiched beam. The system modes and periods based on the boundary conditions and story transfer matrices, found by the shape functions, can then be calculated. Using MATLAB, a computer program was written to solve numerical examples to show the reliability of the method used. Results were compared with previous work done in the literature that was in good agreement. The authors suggested that their method is appropriate to use on a wide range of structural system applications.

CHAPTER III

FINITE ELEMENT THEORETICAL DERIVATIONS

3.1 Introduction

The primary objective of this study is to develop a finite element analysis program coupled with a structural dynamics response modeling program to determine the dynamic response of a structure due to a dynamic load. The use of the finite element method, however, influences the dynamic response of the structure. A variety of finite elements may be chosen for the analysis, each having its advantages and disadvantages. As such, one element is not always superior to another with respect to any given analysis. Often, it is the experience of the analyst that determines the appropriate finite element to be used in the analysis. The purpose of this study is not to determine the appropriateness of the finite element method chosen. In addition, there are also a variety of structural dynamic response algorithms that can be used to determine the response of a structure due to a dynamic load application. As such, it is also not the purpose of this study to determine the “best” algorithm to be used for determining the dynamic response of the structural system. Rather, this study seeks only to combine the theoretical development of the finite element method with an approach for determining the dynamic structural response into a seamless computer application that can be expanded for future use.

To develop the coupled finite element/structural dynamics program, an understanding of the theoretical development of both topics is necessary. Finite elements are the basis for modeling the structural system into a set of discretized pieces that can be assembled into a set of structural equations. The structural dynamics algorithm can then analyze these structural equations at discrete points in time due to a defined dynamic loading. The combination of these two procedures produces displacements of the structural system as a function of time. In the discussion to follow herein, the theoretical development will center on these topics. Included in these discussions will be formulations of element matrices, derivations of finite element method equations, dynamic analysis procedures and solution techniques which were used while developing the coupled finite element analysis/structural dynamics program.

3.2 Interpolation and Shape Functions

The finite element method, at its most basic form, is a set of interpolations or approximations of a dependent variable with respect to an independent variable. Unknown degrees-of-freedom (DOF) are utilized to ensure a single-valued approximation for a set of conditions. An interpolating polynomial with dependent variable ϕ and independent variable x in terms of generalized DOF a_i can be expressed in the form

$$\phi = \sum_{i=0}^n a_i x^i \text{ or } \phi = [X]\{a\} \quad (3-1)$$

where X and a can be written as vectors

$$[X] = [1 \ x \ x^2 \ \dots \ x^n] \text{ and } \{a\} = [a_0 \ a_1 \ a_2 \ \dots \ a_n]^T \quad (3-2)$$

For linear interpolation n can be taken 1 and for quadratic interpolation n can be taken 2.

The a_i can be written in terms of nodal values (known locations of x) of ϕ for known values of x . The relation between a_i and the nodal values can be written as

$$\{\Phi_e\} = [A]\{a\} \quad (3-3)$$

where each row of $[A]$ is $[X]$ calculated at the appropriate nodal location. From the equations above, substitution produces

$$\phi = [N]\{\Phi_e\} \text{ where } [N] = [X][A]^{-1} = [N_1 \ N_2 \ \dots] \quad (3-4)$$

In matrix $[N]$ an individual N_i is represented as a shape function. In this study, the bilinear rectangle (Q4) finite element is considered. Therefore, linear interpolation is considered when generating the shape functions. That is to say that between two points (x_1, ϕ_1) and (x_2, ϕ_2) on a linear line for $[X] = [1 \ x]$ we can obtain

$$\begin{Bmatrix} \phi_1 \\ \phi_2 \end{Bmatrix} = [A] \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} \text{ where } [A] = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix} \quad (3-5)$$

Inverting $[A]$ and using equation 3-4,

$$[A]^{-1} = \frac{1}{x_2 - x_1} \begin{bmatrix} x_2 & x_1 \\ -1 & 1 \end{bmatrix} \text{ and } [N] = \begin{bmatrix} \frac{x_2 - x}{x_2 - x_1} & \frac{x - x_1}{x_2 - x_1} \end{bmatrix} \quad (3-6)$$

Equation 3-6 gives the shape functions of two points on a straight line which are called N_1 and N_2 .

Consider the general Q4 element shown in Figure 3-1. To find the shape functions of the Q4 element, linear interpolation is made along the top and bottom sides to obtain side displacement u_{12} and u_{43} . Thus, in equation 3-5, $x_1 = -a$ and $x_2 = a$, so that

$$u_{12} = \frac{a-x}{2a}u_1 + \frac{a+x}{2a}u_4, u_{43} = \frac{a-x}{2a}u_4 + \frac{a+x}{2a}u_3 \quad (3-7)$$

Linear interpolation is then made in the y direction between u_{12} and u_{43} as

$$u = \frac{b-y}{2b}u_{12} + \frac{b+y}{2b}u_{43} \quad (3-8)$$

Substitution of equations 3-7 into 3-8 yields $u = \sum N_i u_i$ which gives the shape functions of the rectangular Q4 element used in this study as

$$\begin{aligned} N_1 &= \frac{1}{4ab}(1-x)(1-y) \\ N_2 &= \frac{1}{4ab}(1+x)(1-y) \\ N_3 &= \frac{1}{4ab}(1+x)(1+y) \\ N_4 &= \frac{1}{4ab}(1-x)(1+y) \end{aligned} \quad (3-9)$$

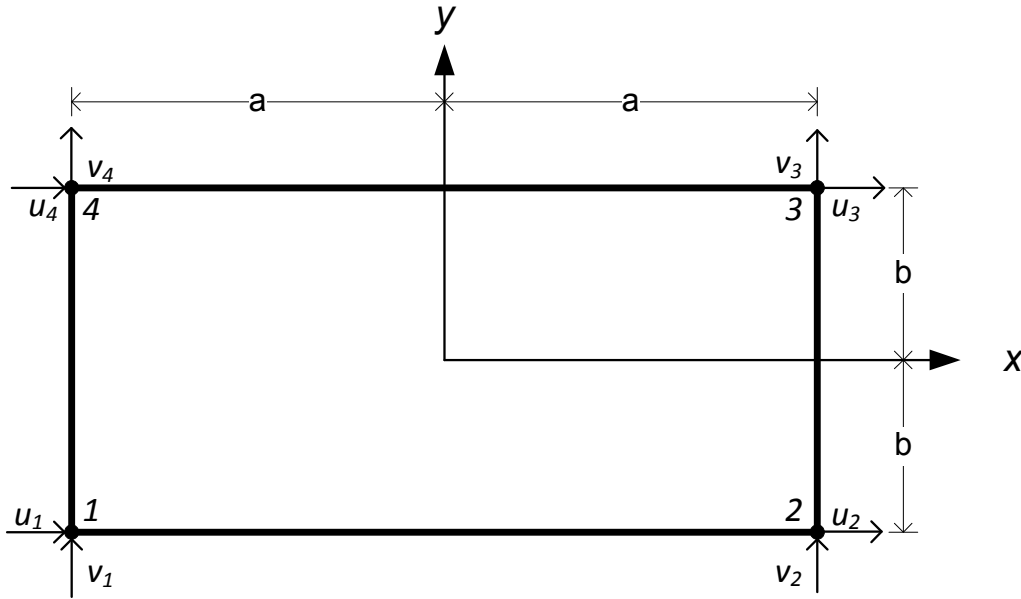


Figure 3-1: Q4 Finite Element Nodal Degrees of Freedom

A similar procedure is repeated for the DOF in the y direction. Namely, these DOF are v_1, v_2, v_3 and v_4 . Similarly, the same shape functions given in equation 3-9 are found for the shape functions in the y-direction. Therefore, the same interpolation, or approximation, is made in both the x (or u) and y (or v) directions.

3.3 Formulation of Element Stiffness Matrices

The principal of virtual work is used to obtain element stiffness matrix for the Q4 element shown in Figure 3-1. This is appropriate for commonly used elements, which are based on interpolation of displacements from nodal DOF. The principal of virtual work states

$$\int \{\delta \varepsilon\}^T \{\sigma\} dV = \int \{\delta u\}^T \{F\} dV + \int \{\delta u\}^T \{\Phi\} dS \quad (3-10)$$

where $\{\delta \varepsilon\}, \{F\}$ and $\{\Phi\}$ represent the virtual strains produced by the virtual displacements, body forces and surface tractions, respectively. The displacements $\{u\}$ are interpolated over an element utilizing shape functions such as those provided by equation 3-9 as

$$\{u\} = [N]\{d\} \quad \text{where} \quad \{u\} = [u \ v \ w]^T \quad (3-11)$$

$$\{\varepsilon\} = [\delta]\{u\} \quad \text{and} \quad \{\varepsilon\} = [B]\{d\} \quad \text{where} \quad [B] = [\delta][N] \quad (3-12)$$

where $[B]$ is the strain-displacement matrix. From equations 3-11 and 3-12

$$\{\delta u\}^T = \{\delta d\}^T [N]^T \quad \text{and} \quad \{\delta \varepsilon\}^T = \{\delta d\}^T [B]^T \quad (3-13)$$

Substitution of equation 3-13 back into the statement of virtual work, equation 3-10, produces

$$\{\delta d\}^T (\int [B]^T [E] [B] dV - \int [B]^T [E] \{e_0\} dV + \int [B]^T \{\sigma_0\} dV - NT F dV - NT \Phi dS = 0 \quad (3-14)$$

Equation 3-14 can be simplified to produce

$$[k]\{d\} = \{r_e\} \quad (3-15)$$

As a result of the principle of virtual work, the element stiffness matrix can be determined as

$$[k] = \int [B]^T [E] [B] dV \quad (3-16)$$

Specifically, for the Q4 element, equation 3-16 can be written as

$$[k] = \int_{-b}^b \int_{-a}^a [B]^T [E] [B] t dx dy \quad (3-17)$$

For the case of two-dimensional plane stress analysis (as considered in this study), the material constitutive matrix [E] is

$$[E] = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \quad (3-18)$$

In two-dimensional analyses, the thickness, t , in equation 3-17 is commonly taken as unity.

3.4 Element Boundary and Loading Conditions

Before the solution of the structural equations, both boundary conditions and loading conditions must be prescribed for the system. Without boundary conditions the structural equations will not produce a single unique solution for the prescribed loading conditions. As such, the structural system will have rigid body motions. Without loading conditions

the structural equations will produce no displacements of the structural system. Thus, it is necessary that both boundary and loading conditions be prescribed for the structural system. These conditions are prescribed at particular DOF of the structural system.

Boundary conditions, or support conditions, can be arranged by providing the appropriate stiffness to the related DOF to produce a prescribed displacement. For zero displacement, the prescribed stiffness can be a numerically large number (several orders of magnitude larger than the largest magnitude in the stiffness matrix) such that a relatively small displacement at that DOF is produced. The boundary conditions can be applied to any DOF on the structure no matter its direction. In the present study, only translational DOF are considered in the finite elements (i.e., there are no rotational DOF) and therefore only translational displacements are restricted with respect to particular support conditions such as a fixed-support, pinned-support, or roller-support.

Loading conditions are prescribed in a fashion similar to boundary/support conditions. The load can be applied to the any DOF. The structural system may be subjected to a single loading condition or multiple loading conditions. As this study considers dynamic loads, beyond the typical static load cases considered in a typical finite element analysis, further explanation of the dynamic load is presented in Section 3.6.3.

3.5 Solution Techniques

Equation 3-17 is integrated over a rectangular surface as given in Figure 3-1. Numerically, however, to integrate this equation is cumbersome and not efficient. Thus, integration is achieved using Gauss Quadrature. Gauss Quadrature is a numerical approximation of the integration by use of simple algebraic equations evaluated at specific points. To use Gauss Quadrature, however, the element must be formulated in the isoparametric space. The use of the isoparametric space specifies that an isoparametric element is used rather than the physical element. Thus, the integrands in the integration formulas are expressed as functions of ξ and η rather than x and y . For the function $\phi = \phi(\xi, \eta)$, the Quadrature rule is given as

$$I = \int_{-1}^1 \int_{-1}^1 \phi(\xi, \eta) d\xi d\eta = \sum_i \sum_j W_i W_j \phi(\xi, \eta) \quad (3-19)$$

where W_i and W_j are weighting factors for each Gauss point i and j . The weighting factors for two-point Gauss Quadrature, as used in this study for the Q4 element, are taken as unity.

For the Q4 element shown in Figure 3-1, the individual isoparametric shape functions can simply be determined from equation 3-9 by assigning $a = 1$, $b = 1$, $x = \xi$ and $y = \eta$ producing

$$\begin{aligned} N_1 &= \frac{1}{4}(1 - \xi)(1 - \eta) \\ N_2 &= \frac{1}{4}(1 + \xi)(1 - \eta) \\ N_3 &= \frac{1}{4}(1 + \xi)(1 + \eta) \\ N_4 &= \frac{1}{4}(1 - \xi)(1 + \eta) \end{aligned} \quad (3-20)$$

Since the isoparametric Q4 element is not in the same coordinate system as the physical Q4 element, a mapping function is needed to relate the two coordinate systems. The Jacobian matrix is used to accomplish this mapping. The Jacobian matrix is a scale factor that multiplies $d\xi d\eta$ to produce the physical area increment $dx dy$ and is expressed as (for the Q4 isoparametric element)

$$[J] = \frac{1}{4} \begin{bmatrix} -(1-\eta) & (1-\eta) & (1+\eta) & -(1+\eta) \\ -(1-\xi) & -(1+\xi) & (1+\xi) & (1-\xi) \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \quad (3-21)$$

Thus, the element stiffness matrix for the Q4 isoparametric element is then

$$[k] = \iint [B]^T [E] [B] t \, dx \, dy = \int_{-1}^1 \int_{-1}^1 [B]^T [E] [B] t J \, d\xi \, d\eta \quad (3-22)$$

By substituting equation 3-22 into equation 3-19, the integration can be numerically calculated using Gauss Quadrature.

Following the determination of the stiffness of each element in the finite element mesh, the system stiffness must be assembled. Based on the arrangement of the finite element mesh, the stiffness for each element corresponding to particular system DOF is “fed” into the system stiffness corresponding to the same system DOF. This process, called the assembly process, is simply a mapping technique relating the DOF corresponding to each element to the DOF of the system.

After assembly of the elements into the system equations, the structural equations are then in the form that can be solved. Gauss elimination is used to solve the structural equations for given boundary and loading conditions. In Gauss elimination, equations

$[K]\{D\} = \{R\}$ are solved for $\{D\}$, the displacements, by reducing $[K]$ to upper triangular form and then solving for unknowns in the reverse order by back substitution.

3.6 Structural Dynamics

An integral part of this study is the incorporation of a structural dynamics response algorithm into the finite element method. These two coupled methods will produce a seamless method by which to analyze a structural system, using the finite element method, under an applied dynamic load(s) for a specified period of time. As a result, the response of the structural system as a function of time will be determined. The following sections explain the theoretical development of the structural dynamics methodology used in this study. Structural dynamics derivations and solution techniques are outlined briefly. The equations are derived for time-dependent loads. This study only considers undamped structures, thus, the formulation of damping is not presented and it is taken as zero in all equations. The following sections introduce the theoretical explanations of structural dynamics that were used in the development of the coupled computer program.

3.6.1 Equation of Motion

The equation of motion is the basic and fundamental part of structural dynamics. All formulations are derived based on the equation of motion in structural dynamics. The equation of motion is generally given as

$$m\ddot{u} + c\dot{u} + ku = p(t) \quad (3-23)$$

The right side of the equation is the time dependent force, $p(t)$. On the left side of equation 3-23, m represents mass, c represents damping, k represents stiffness and u , \dot{u} , \ddot{u} represent displacement, velocity and acceleration, respectively (the overdot represents a time derivative). The structure system, which has multiple DOFs, is of course considered in this study. The specific number of DOF depends on the finite element model of the structural system. As a result, all variables in equation 3-23 are matrices or column vectors of size related to the number of DOF for the system.

3.6.2 Formulation of Element Mass Matrices

The formulation of the element mass matrix is based on the virtual work principal and is similar to the formulation of the element stiffness matrix as discussed in Section 3.3. The work done by externally applied loads is equal to the sum of the work absorbed by inertial, dissipative, and internal forces for any virtual displacement. For an element volume V and surface S

$$\begin{aligned} \int \{\delta u\}^T \{F\} dV + \int \{\delta u\}^T \{\Phi\} dS + \sum_{i=1}^n \{\delta u\}_i^T \{p\}_i \\ = \int (\{\delta u\}^T \rho \{\ddot{u}\} + \{\delta u\}^T c \{\dot{u}\} + \{\delta \varepsilon\}^T \{\sigma\}) dV \end{aligned} \quad (3-24)$$

Where $\{F\}$ and $\{\Phi\}$ are prescribed body forces and surface tractions, $\{p\}_i$ and $\{\delta u\}_i$ are prescribed concentrated loads and their corresponding virtual displacements, ρ is mass density and c is the damping coefficient. Following Section 3.3, the nodal displacements, nodal velocities, nodal accelerations, and strains are approximated by

$$\{u\} = [N]\{d\} \quad \{\dot{u}\} = [N]\{\dot{d}\} \quad \{\ddot{u}\} = [N]\{\ddot{d}\} \quad \{\varepsilon\} = [B]\{d\} \quad (3-25)$$

Substitution of equation 3-25 into equation 3-24 produces the virtual work expression

$$\{\delta d\}^T \left(\int \rho [N]^T [N] dV \{\ddot{d}\} + \int c [N]^T [N] dV \{\dot{d}\} + \int [B]^T \{\sigma\} dV - \right. \\ \left. N T F dV - N T \Phi dS - i=1 n p i=0 \right) \quad (3-26)$$

The first integral in equation 3-26 provides the element mass matrix as

$$[m] = \int \rho [N]^T [N] dV \quad (3-27)$$

For two-dimensional analysis, equation 3-27 yields

$$[m] = \int_{-b}^b \int_{-a}^a \rho [N]^T [N] t dx dy \quad (3-28)$$

Following the procedure for assembly of element stiffness matrix into the system matrix, the element mass matrix for each individual element is assembled in the same procedure. The DOF for the mass and stiffness matrices are identical. Thus, from a numerical perspective, the element stiffness and mass matrices can be assembled into the system stiffness and mass matrices simultaneously.

3.6.3 Dynamic Loading Conditions

There are many types of dynamic loads from a realistic perspective. In this study, several common idealized dynamic loadings are considered. Namely, these include the harmonic loading, the impulse loading and the step loading. It should be noted, however, that the MATLAB program written as a consequence of this study can handle any user-input for the dynamic loading. The user only has to provide the value of the load at specified time intervals, or sampling points. These three idealized loads are only chosen based on their commonality in structural dynamic simplifications.

Harmonic loading is a function of the sine or cosine functions and its equation is in general

$$p(t) = p_0 \sin \omega t \text{ or } p_0 \cos \omega t \quad (3-29)$$

where p_0 is the amplitude or maximum value of the force and its frequency ω is the exciting frequency or forcing frequency. Figure 3-2 provides an example of a harmonic loading. Examples of a harmonic loading includes wind driven loading on structures, earthquake (highly idealized) loading, and vehicular motion on bridges.

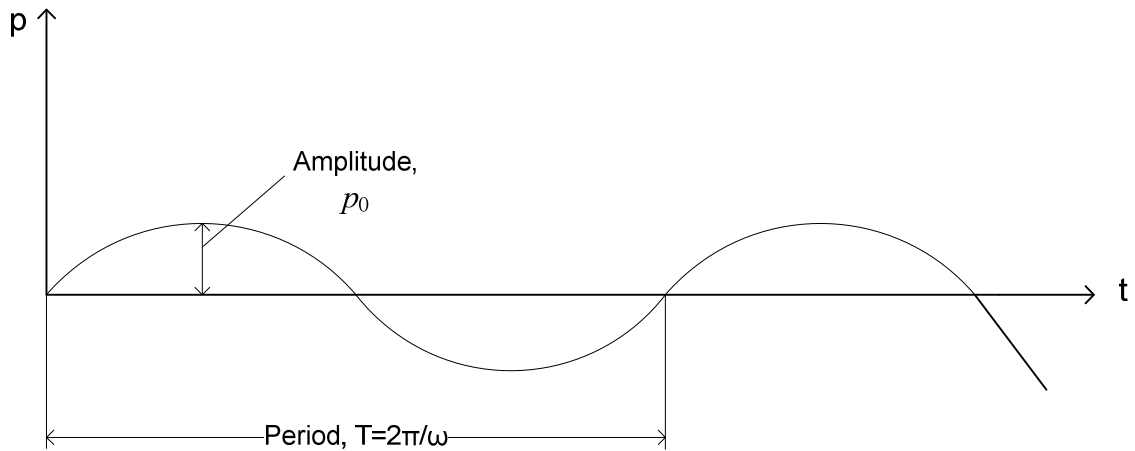


Figure 3-2: Harmonic Loading

A very large force that acts for a very short time but with a time integral that is finite is called an impulsive force. In general, an impulsive force is defined by

$$p(t) = 1/\varepsilon \quad (3-30)$$

with a time duration ε starting at the time instant $t = \tau$, also called the time lag. Figure 3-3 is an illustration of an impulse loading. Examples of an impulsive load primarily include blast loadings such as those due to detonation of blast devices.

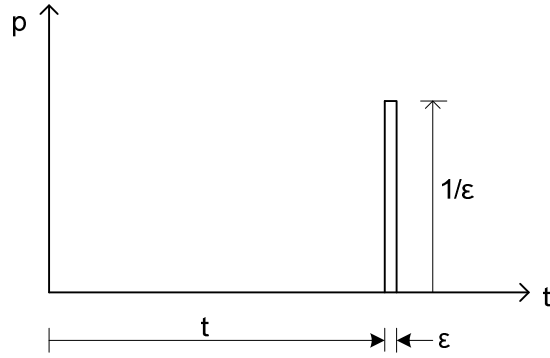


Figure 3-3: Impulse Loading

Finally, another typical dynamic loading is a force that jumps suddenly from zero to magnitude p_0 and stays constant at that value is called step force. In general, the step loading is defined by

$$p(t) = p_0 \quad (3-31)$$

Figure 3-4 provides an example of an impulse loading that jumps to its magnitude p_0 at time 0 seconds as indicated. Examples of a step loading include the sudden application of a full load rather than being applied gradually over time.

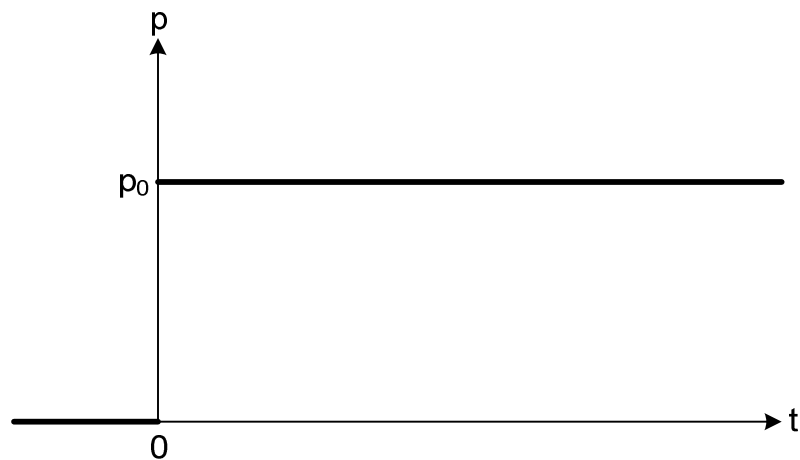


Figure 3-4: Step Loading

3.6.4 Numerical Integration

There are several numerical integration methods for linear systems to numerically integrate the equation of motion previously defined as equation 3-23. In all methods for the purposes of this study, initial displacements and velocities are taken zero (the system is initially at rest in an underformed position) and $p(t)$ is known at all time intervals t . The aim of numerical integration is analyzing the system over time intervals Δt . Some of the numerical integration methods, that are called time-stepping methods, are interpolation of excitation, central difference method and Newmark's method.

Interpolation of excitation uses recurrence formulas. It can be used for small Δt and for linear systems. In addition, this method is suitable for single DOF systems but it is not appropriate for multi DOF systems. Central difference methods are based on finite difference approximations of the time derivatives of displacement, velocity and acceleration. Solution at u_{i+1} which represents the displacement at the step $i+1$ is determined from the equation of motion at time step i . Furthermore, u_i and u_{i-1} must be known to find the displacement at time step $i+1$, u_{i+1} . Newmark's method, used in this study, is a family of time-stepping methods based on the iteration of \ddot{u}_{i+1} . A detailed discussion is presented in the next section.

3.6.5 Newmark-Beta Solution Technique

Newmark's technique, also known as the more general Newmark-Beta solution technique, is based on the following equations

$$\dot{u}_{i+1} = \dot{u}_i + [(1 - \gamma)\Delta t]\ddot{u}_i + (\gamma\Delta t)\ddot{u}_{i+1} \quad (3-32)$$

$$u_{i+1} = u_i + (\Delta t)\dot{u}_i + [(0.5 - \beta)(\Delta t)^2]\ddot{u}_i + [\beta(\Delta t)^2]\ddot{u}_{i+1} \quad (3-33)$$

The parameters β and γ define the variation of acceleration over a time step and determine the stability and accuracy characteristics of the technique. Typical selection of γ is 1/2, and β can vary between 1/4 and 1/6. For the average acceleration method, used in this study, β is taken as 1/4. All equations are matrix equations for multi DOF as is the case in this study. The Newmark-Beta solution procedure can be presented step by step for linear, multi DOF systems.

Initial conditions are defined as $u_0 = u(0)$ and $\dot{u}_0 = \dot{u}(0)$ representing the initial displacement and initial velocity, respectively. Initial calculations are only calculated one time and they are presented in equations 3-34 thorough 3-37 as

Solving for the initial acceleration \ddot{u}_0 ;

$$m\ddot{u}_0 = p_0 - c\dot{u}_0 - ku_0 \quad (3-34)$$

After selecting Δt and the Newmark-Beta parameters β and γ the effective stiffness of the system can be calculated as

$$\hat{k} = k + \frac{\gamma}{\beta\Delta t}c + \frac{1}{\beta\Delta t^2}m \quad (3-35)$$

Calculation constants, a and b , for use at each time step are calculated as

$$a = \frac{1}{\beta \Delta t} m + \frac{\gamma}{\beta} c \quad (3-36)$$

$$b = \frac{1}{2\beta} m + \Delta t \left(\frac{\gamma}{2\beta} - 1 \right) c \quad (3-37)$$

For each time step, equations 3-38 thorough 3-42 are be repeated until all time steps are done. The effective force at time step i is then calculated as

$$\Delta \hat{p} = \Delta p_i + a \dot{u}_i + b \ddot{u}_i \quad (3-38)$$

Solving for the change in displacement during the time step Δu_i is then

$$\hat{k} \Delta u_i = \Delta \hat{p}_i \quad (3-39)$$

During the same time step, the change in velocity and the change in acceleration are also

$$\Delta \dot{u}_i = \frac{\gamma}{\beta \Delta t} \Delta u_i - \frac{\gamma}{\beta} \dot{u}_i + \Delta t \left(1 - \frac{\gamma}{2\beta} \right) \ddot{u}_i \quad (3-40)$$

$$\Delta \ddot{u}_i = \frac{1}{\beta \Delta t^2} \Delta u_i - \frac{1}{\beta \Delta t} \dot{u}_i - \frac{1}{2\beta} \ddot{u}_i \quad (3-41)$$

Finally, updating the displacement, velocity and acceleration can be found relative to the previous position, velocity and acceleration as

$$\begin{aligned} u_{i+1} &= u_i + \Delta u_i \\ \dot{u}_{i+1} &= \dot{u}_i + \Delta \dot{u}_i \\ \ddot{u}_{i+1} &= \ddot{u}_i + \Delta \ddot{u}_i \end{aligned} \quad (3-42)$$

Repeating these calculations, equations 3-38 through 3-42, the displacements, velocities and accelerations can be found for discrete time points for the structural system.

CHAPTER IV

CASE STUDIES

4.1 Introduction

Case studies are provided to illustrate the capabilities of the MATLAB FEA program. This code has one main program file, one input file and multiple subroutines to analyze the structural system. Material properties, geometrical properties, boundary conditions, loading conditions and more can be easily defined by the user in the input file of the MATLAB code. The case studies all consider two-dimensional dynamically loaded beams modeled using a finite element mesh of Q4 elements. To confirm the procedure and accuracy, a basic example is solved by hand in Appendix B of this study. Additionally, the main program and input file of the MATLAB code is provided in Appendix A of this study.

Table 4-1 shows the beams properties that are used in the four case studies. E represents modulus of elasticity; ν and ρ represent Poisson's ratio and the mass density of the beam, respectively. In the first case study, the beam is fixed at the left end. The second case study considers a beam that is simply supported. Finally, the third and fourth case studies both consider continuous beams that are supported at the left end, the right end, and at the

midpoint. The geometry of the beams, boundary conditions and loading conditions are provided with the case studies.

	Modulus of Elasticity (psi)	Poisson's Ratio	Mass Density (lbs/in ³)	Number of Elements	Number of Nodes	Type of Loading	Beam Length (in)	Beam Depth (in)
Case 1	4.35(10 ⁶)	0.15	0.0868	48	65	Harmonic	144	12
Case 2	4.35(10 ⁶)	0.15	0.0868	48	65	Impulse	144	12
Case 3	4.35(10 ⁶)	0.15	0.0868	57	80	Step	240	12
Case 4	4.35(10 ⁶)	0.15	0.0868	57	80	Multiple	240	12

Table 4-1: Case Studies

As seen in Table 4-1, four types of loading cases are considered. The system is undamped for all cases. Because two-dimensional analyses are considered, each node has two degrees of freedom in the x and y directions while the beam width is taken as unity for all cases.

The dynamic analysis duration is 10 seconds, the time step Δt is taken as 0.1 seconds and initial displacements and velocities for all DOF are zero at time = 0 seconds. With the given time duration and time step, each case study has 100 time steps to solve in the dynamic response analysis. The MATLAB FEA program has the capability to produce movies of the displacements during the time duration of the dynamic loading. As it is not possible to place a movie in a text format, select “snap-shots” have been provided for each case study to illustrate the dynamic response of the structural system. These figures serve only as illustrations as to the full capabilities of the MATLAB FEA program developed as a consequence of this study.

4.2 Harmonic Loading

A harmonic loading is applied to the first beam which is cantilever beam. The illustration of the beam is given in the Figure 4-1. The applied load F represents the harmonic load as it is applied at the right side of the beam. The properties of the beam are provided in Table 4-1 as Case 1.

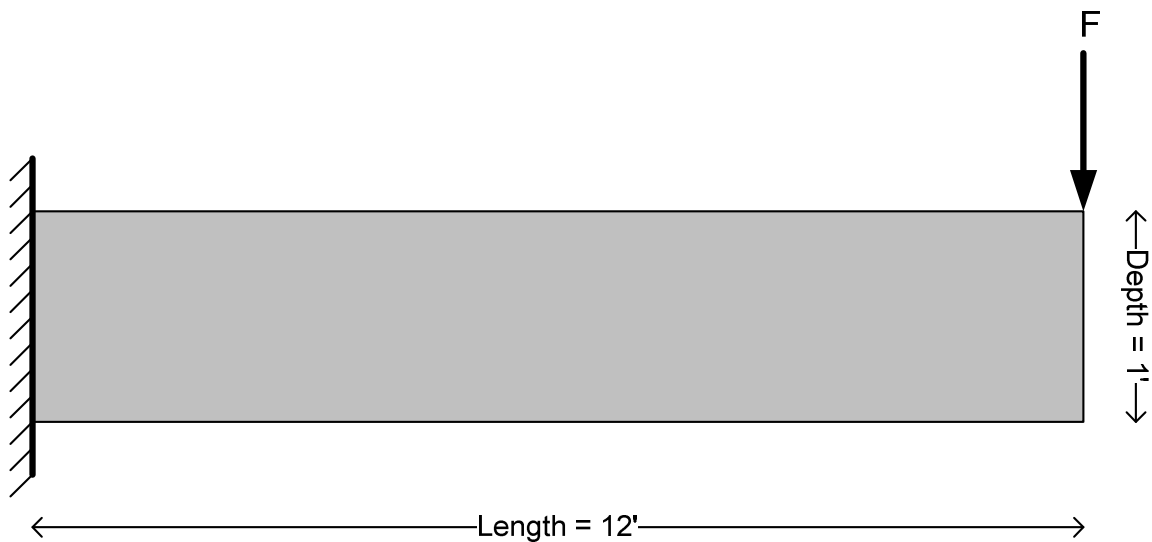


Figure 4-1: Case Study 1 – Cantilever Beam

The harmonic load is a sine function and is illustrated in Figure 4-2. The harmonic load is applied between 0 and 10 seconds with sampling points taken at intervals of 0.1 seconds. The magnitude of the load varies between 1800 lbs and -1800 lbs.

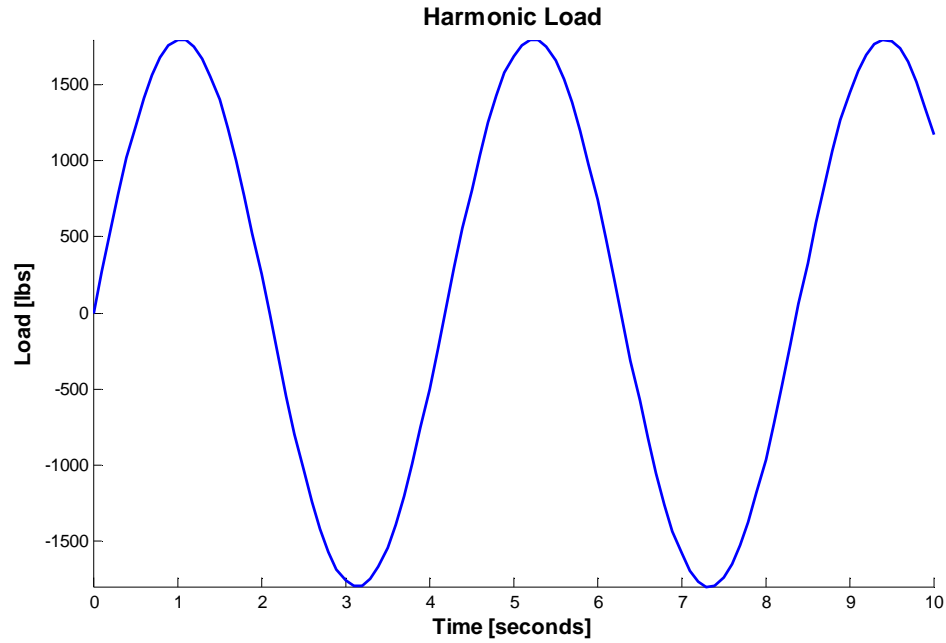


Figure 4-2: Case Study 1 – Harmonic Loading

As stated in Table 4-1, the beam is formulated with 48 rectangular Q4 elements having 130 DOFs. Illustration of the finite element mesh of the beam is shown in Figure 4-3. Green circles indicate node numbering and red circles indicate element numbering in the mesh. Figure 4-4 illustrates the displacement of the beam considering a mesh of Q4 finite elements. Sub-figures (a) – (f) provide a representation of the movie generated at discrete time intervals as indicated on each sub-figure.

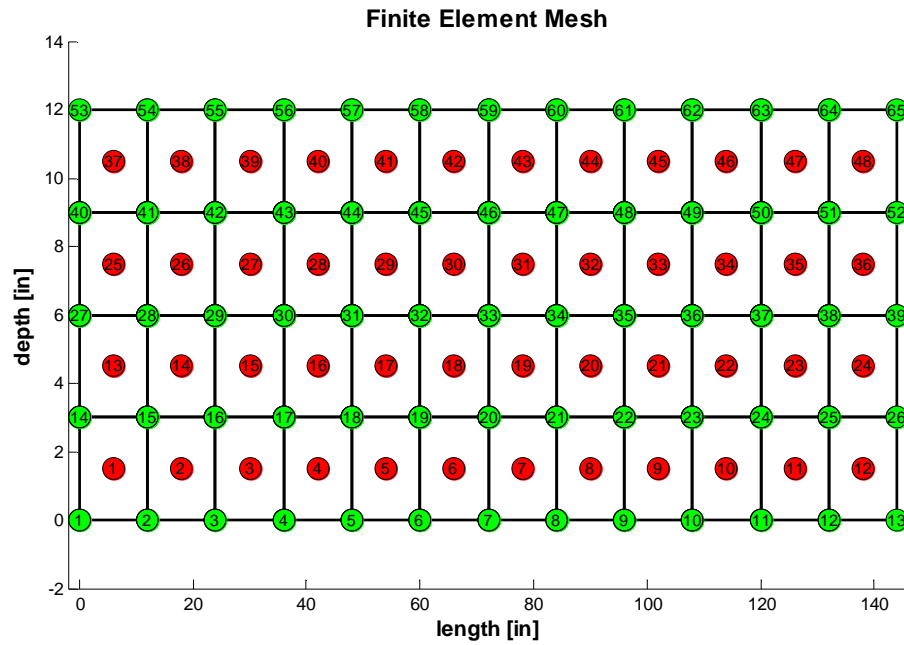


Figure 4-3: Case Study 1 – Finite Element Mesh

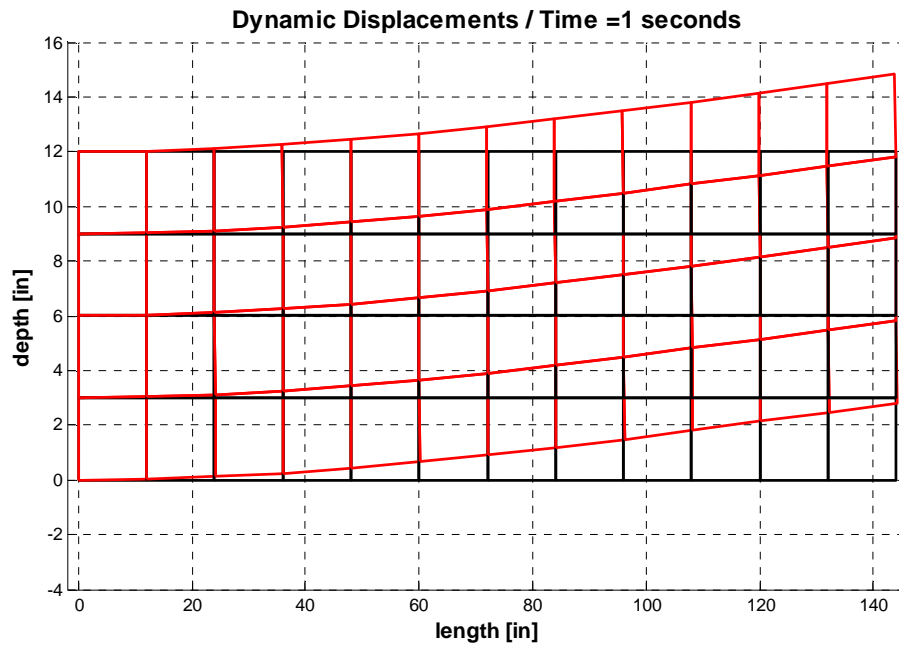


Figure 4-4(a): Case Study 1 – Displacements

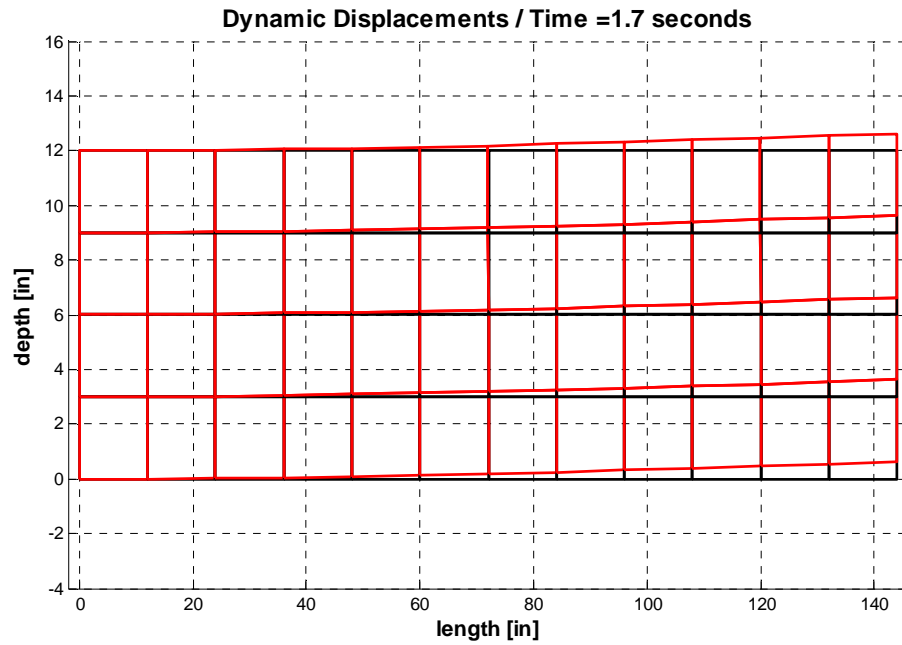


Figure 4-4(b): Case Study 1 – Displacements

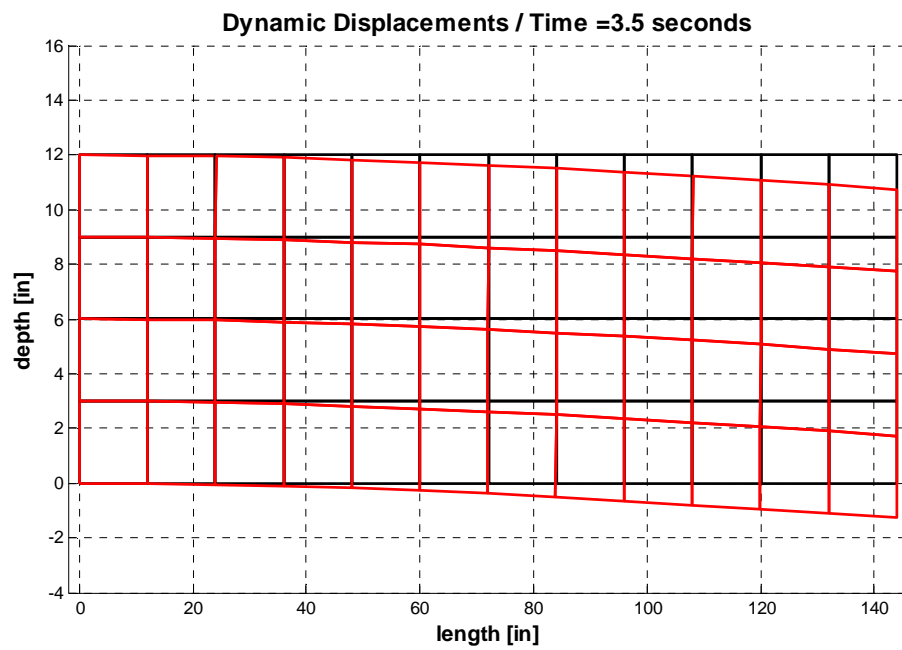


Figure 4-4(c): Case Study 1 – Displacements

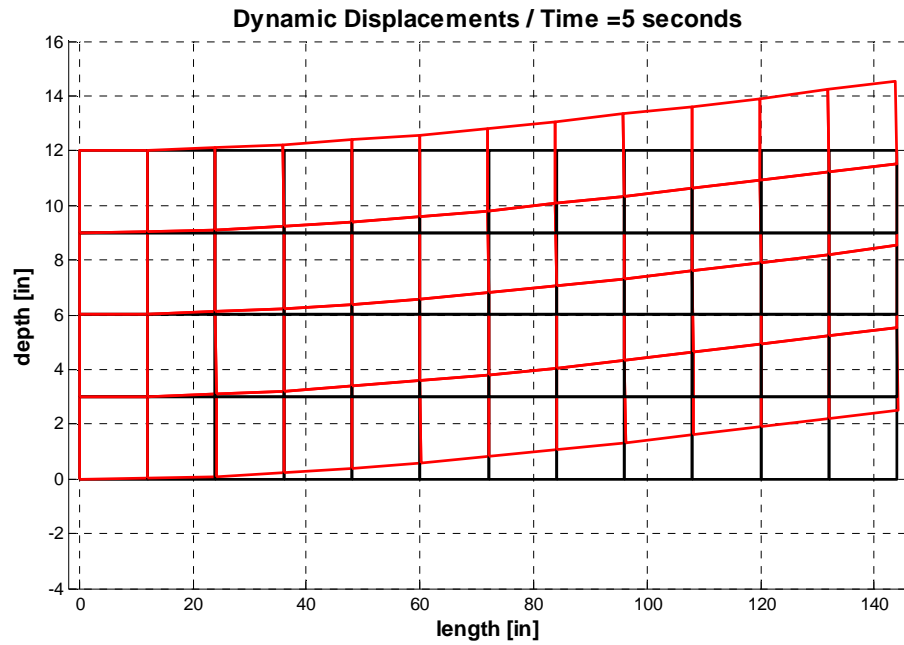


Figure 4-4(d): Case Study 1 – Displacements

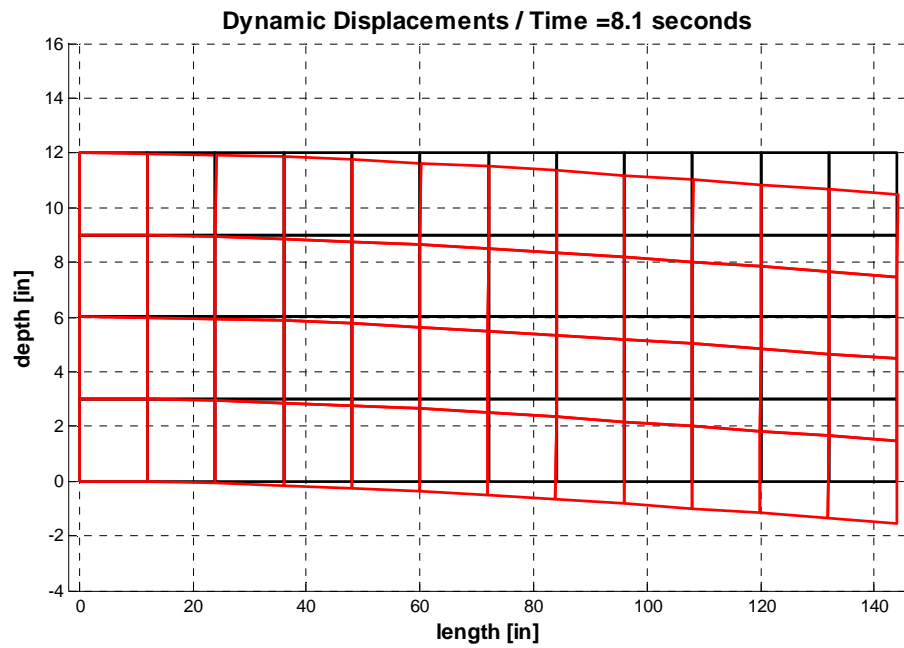


Figure 4-4(e): Case Study 1 – Displacements

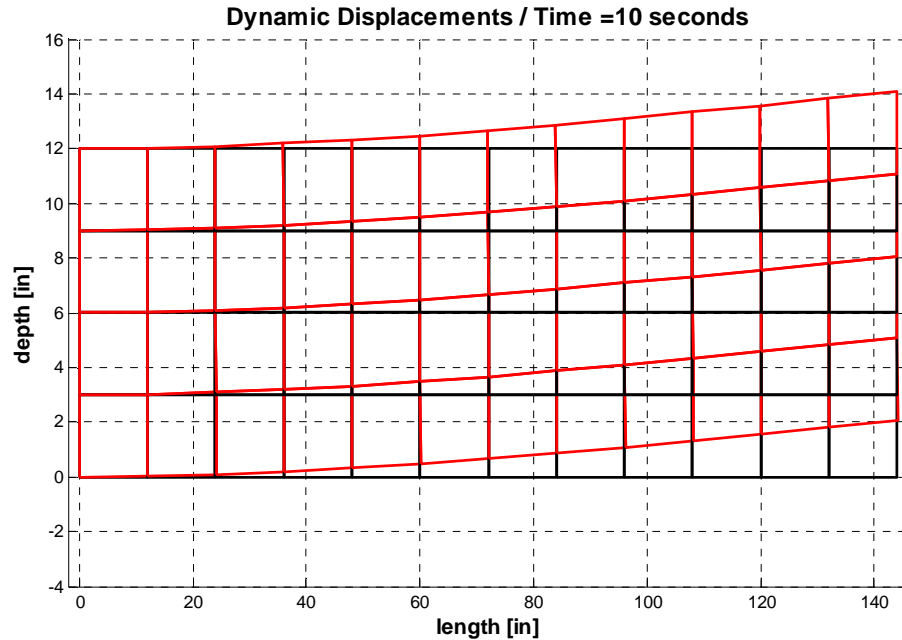


Figure 4-4(f): Case Study 1 – Displacements

As seen from the sub-figures of Figure 4-4, the free end of the beam moves up and down due to the harmonic loading. Comparison of hand calculations of the steady-state displacement response and computer results are presented in Appendix B. DOF 78 at node 39 is considered while comparing the results of displacements in Appendix B.

4.3 Impulse Loading

An impulse loading is applied to the 12ft long simply-supported beam. The load is applied at the middle of the beam. The illustration of the beam is shown in Figure 4-5. The applied load F represents the impulse load in the figure. The properties of the beam are provided in Table 4-1 as Case 2.

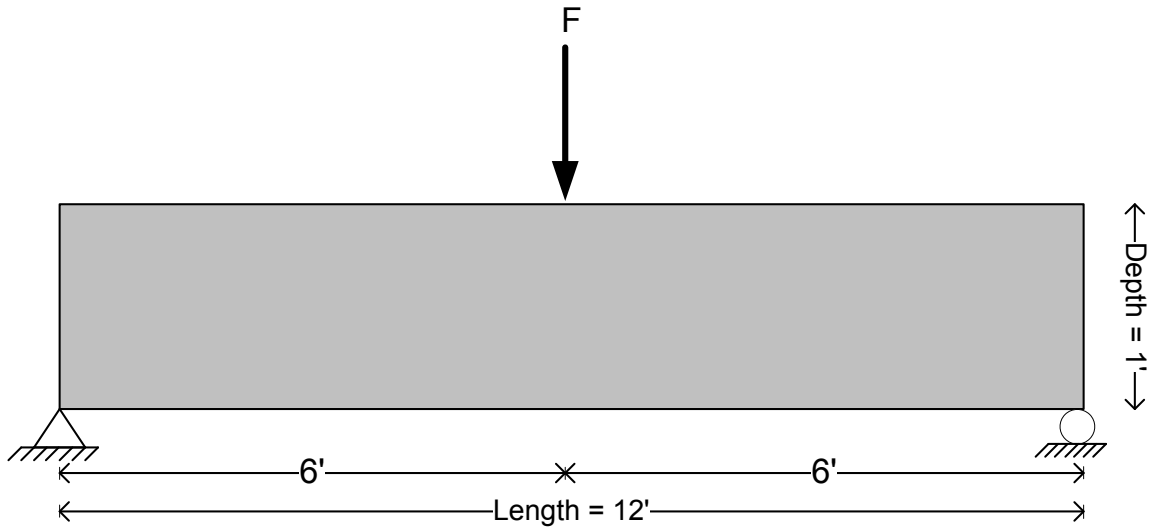


Figure 4-5: Case Study 2 – Simply-Supported Beam

The impulse loading, as shown in Figure 4-6, is applied between 0 and 10 seconds with sampling points taken at intervals of 0.1 seconds. At time = 1s the impulse loading immediately jumps to a value of -2500 lbs. At all other times t , the loading is zero.

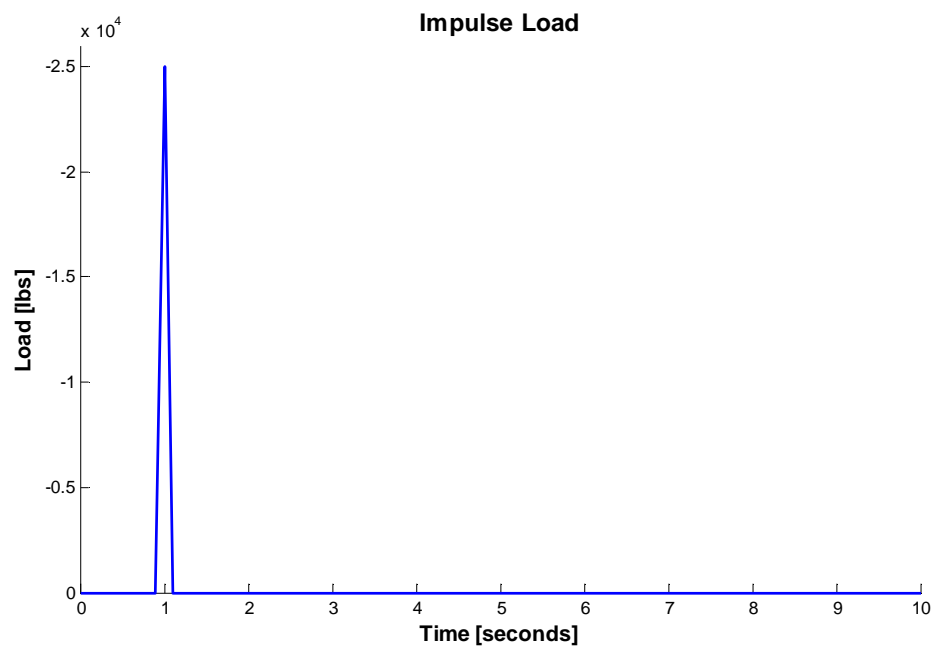


Figure 4-6: Case Study 2 – Impulse Loading

As stated in Table 4-1, the beam, which has 130 DOFs, is constructed with 48 rectangular Q4 elements. Illustration of the finite element mesh of the beam is shown in Figure 4-7. Figure 4-8 illustrates the displacement of the beam considering a mesh of Q4 finite elements. Sub-figures (a) – (f) provide a representation of the movie generated at discrete time intervals as indicated on each sub-figure.

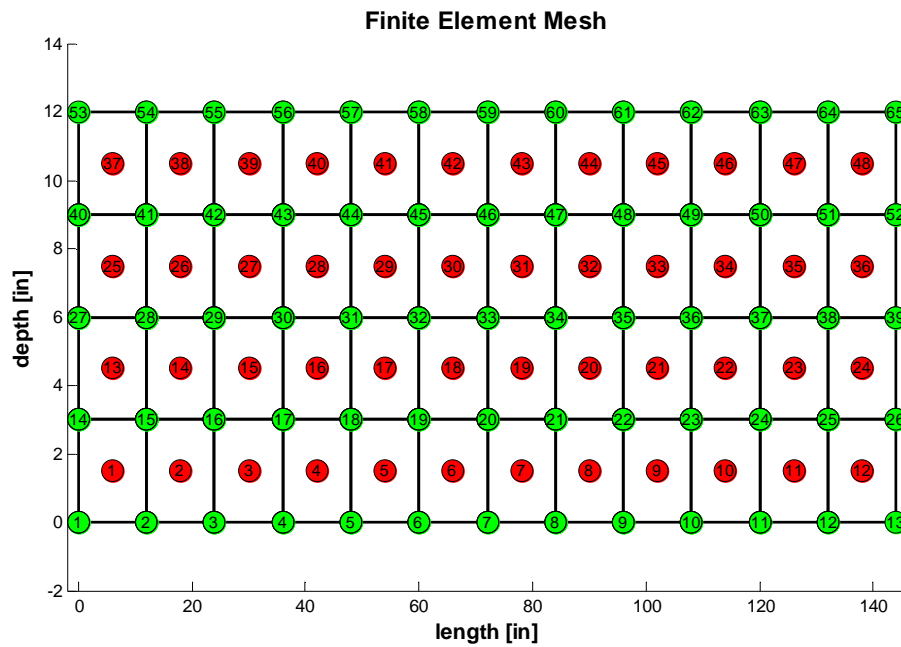


Figure 4-7: Case Study 2 – Finite Element Mesh

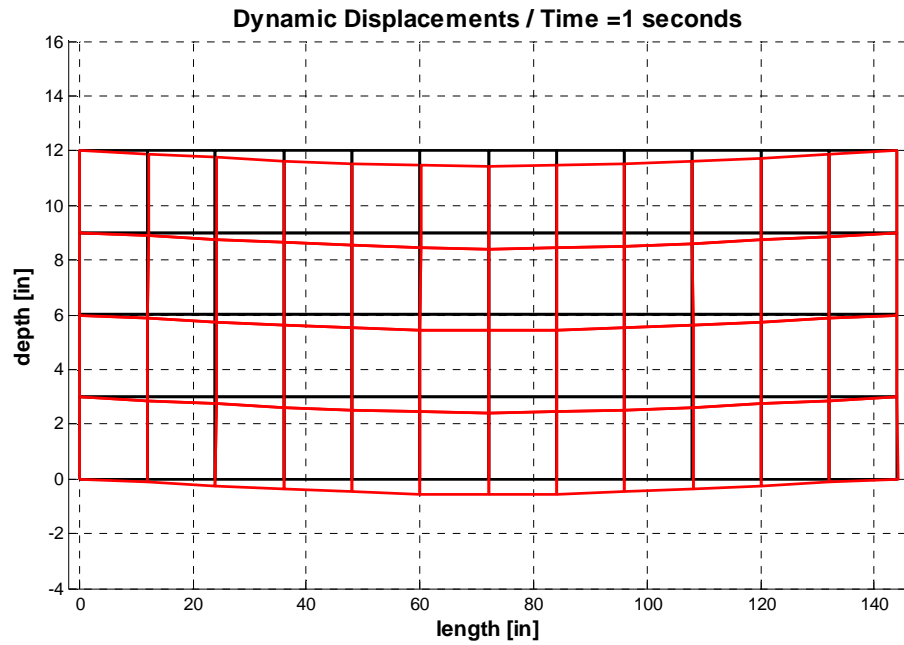


Figure 4-8(a): Case Study 2 – Displacements

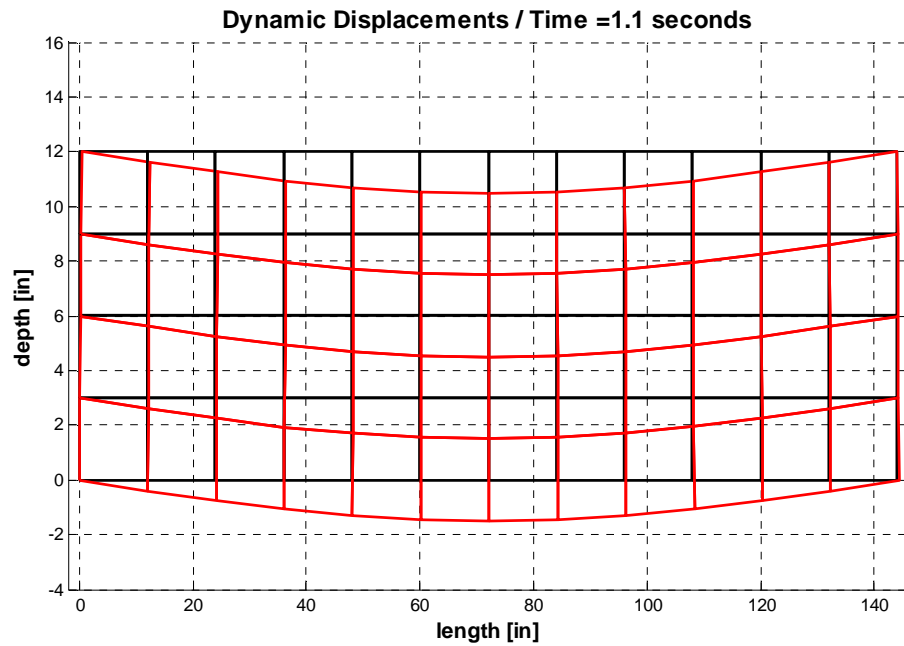


Figure 4-8(b): Case Study 2 – Displacements

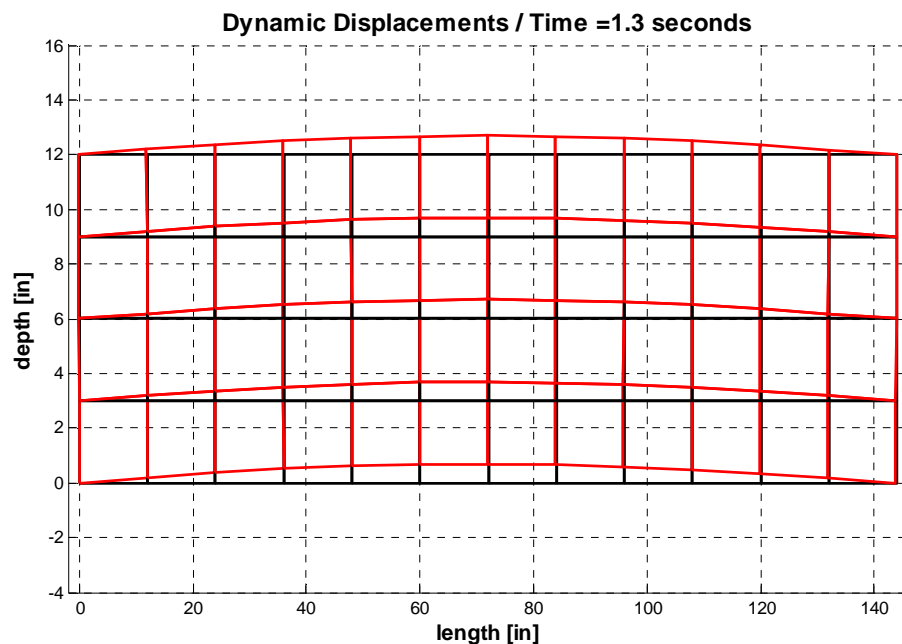


Figure 4-8 (c): Case Study 2 – Displacements

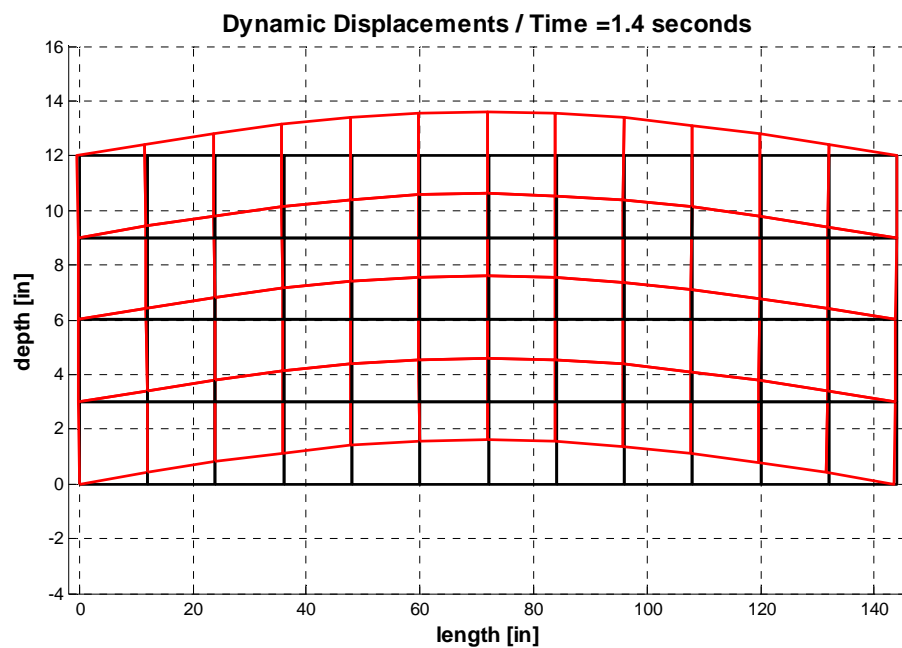


Figure 4-8(d): Case Study 2 – Displacements

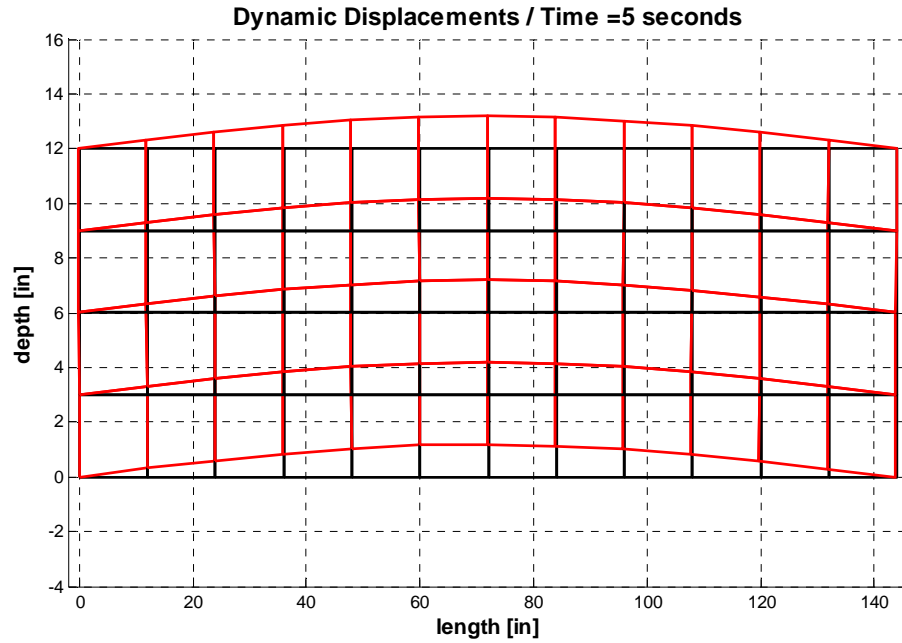


Figure 4-8(e): Case Study 2 – Displacements

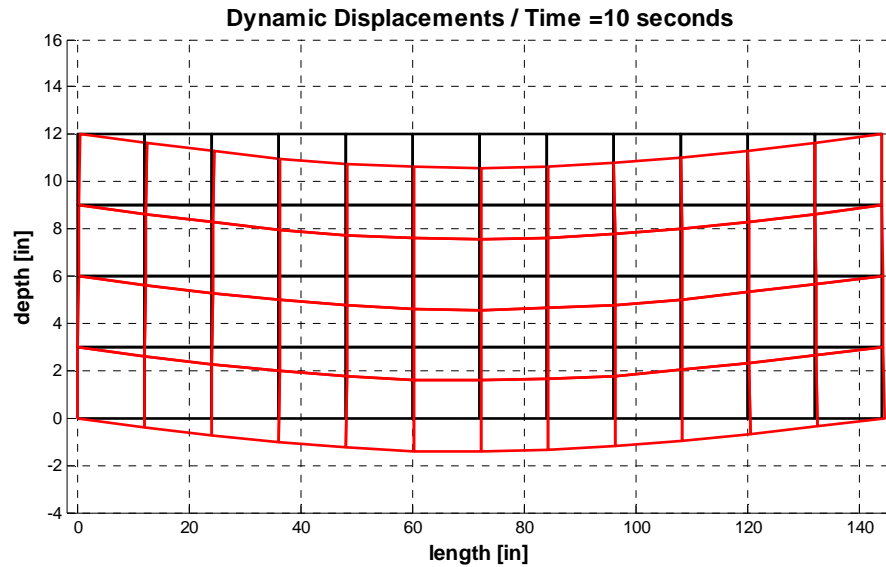


Figure 4-8(f): Case Study 2 – Displacements

Until time = 1 second there is no displacement at the beam as expected. At that time, the impulse load is applied to the beam. Because the system is undamped, after 1 second the

beam starts oscillating up and down and goes forever with the maximum displacement at the midpoint as seen from the Figure 4-8 and its subfigures.

4.4 Step Loading

A step loading is applied to the third case study. The 20ft long beam has a pin support at the left end and roller supports at the middle and right end. The load is located 5 feet from the right side of the beam. The illustration of the beam is shown in Figure 4-9. The applied load F represents the step load in the figure. The properties of the beam are provided in Table 4-1 as Case 3.

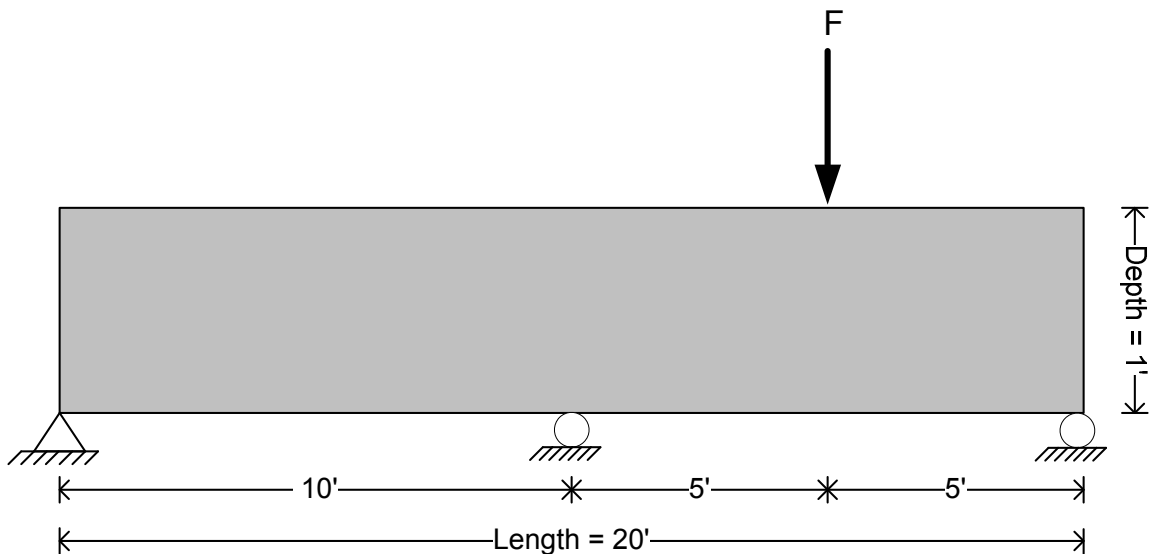


Figure 4-9: Case Study 3 – Continuous Beam

The step loading, as shown in Figure 4-10, is applied between 0 and 10 seconds with sampling points taken at intervals of 0.1 seconds. As with the impulse loading of Case 2,

there is no load applied before the time reaches 1 second. At this point in time, the value of the load jumps suddenly to -20000 lbs and stays constant for the duration of time.

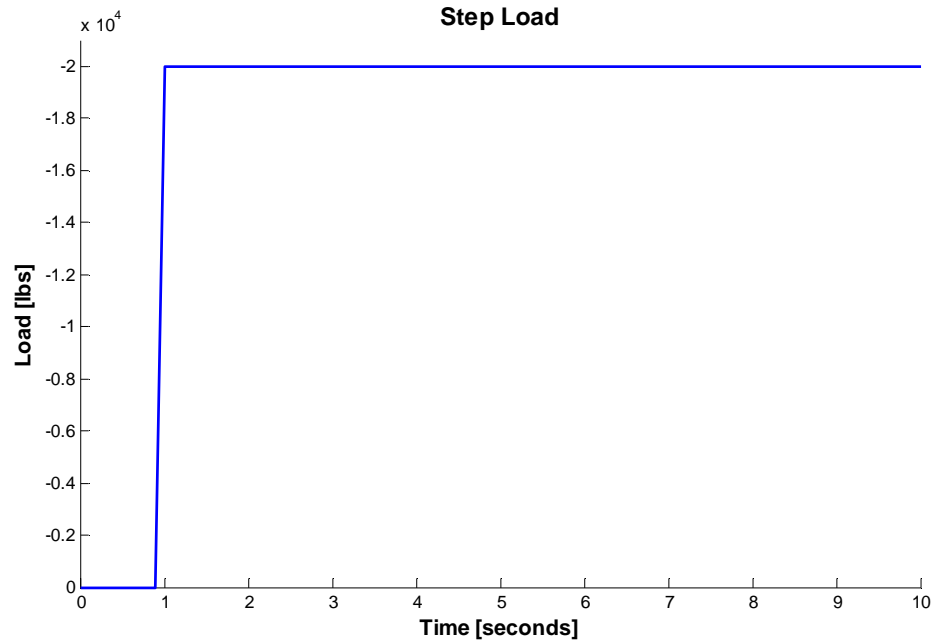


Figure 4-10: Case Study 3 – Step Loading

As stated in Table 4-1, the beam is comprised of 57 rectangular Q4 elements having 160 DOFs. Illustration of the finite element mesh of the beam is shown in Figure 4-11. Figure 4-12 illustrates the displacement of the beam considering a mesh of Q4 finite elements. Sub-figures (a) – (f) provide a representation of the movie generated at discrete time intervals as indicated on each sub-figure.

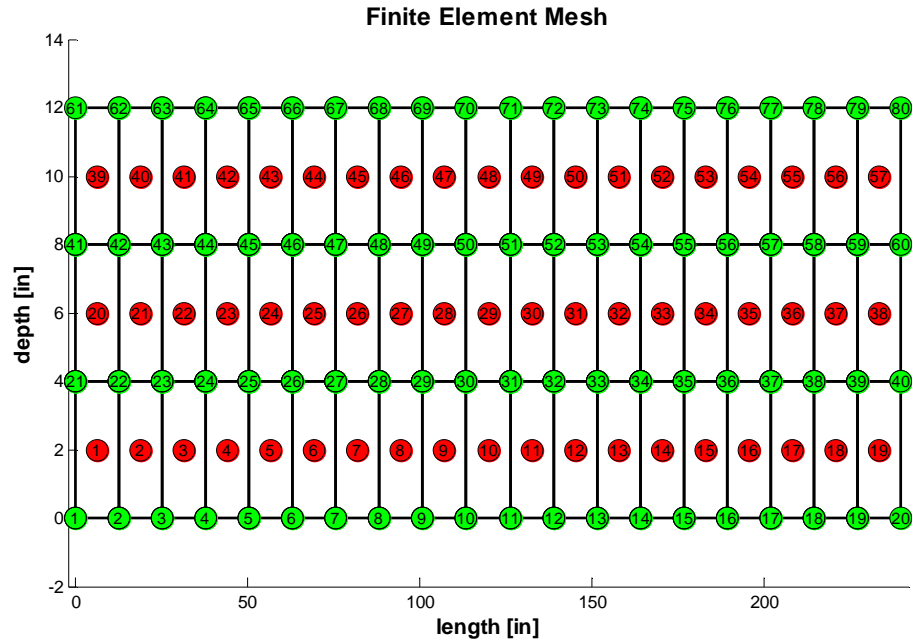


Figure 4-11: Case Study 3 – Finite Element Mesh

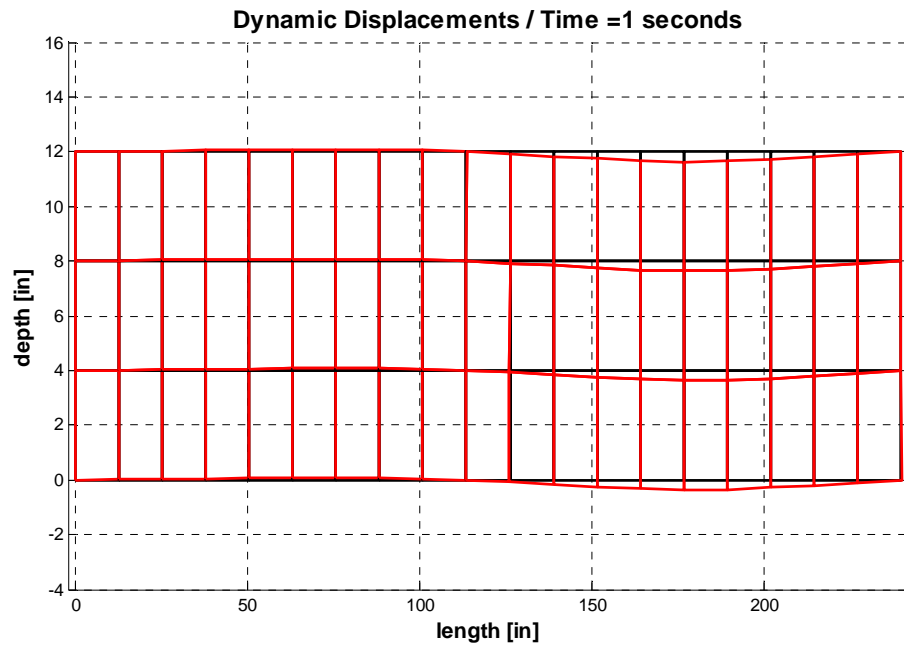


Figure 4-12(a): Case Study 3 – Displacements

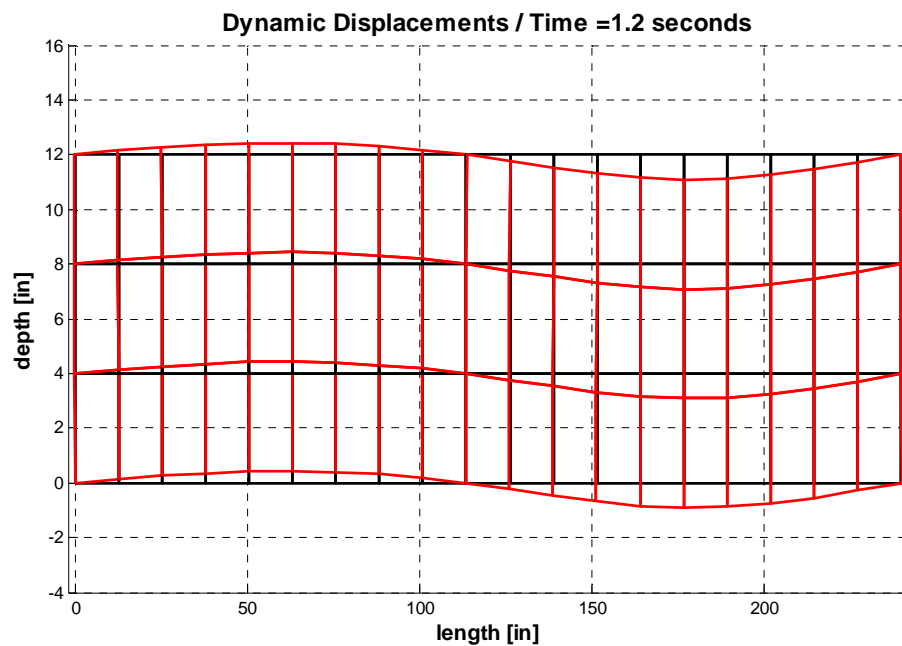


Figure 4-12(b): Case Study 3 – Displacements

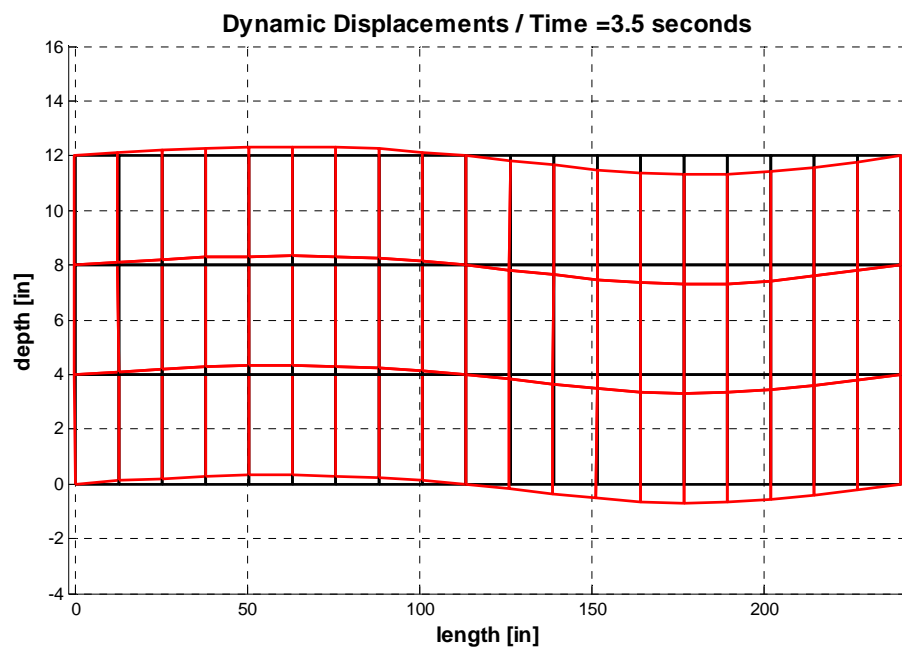


Figure 4-12 (c): Case Study 3 – Displacements

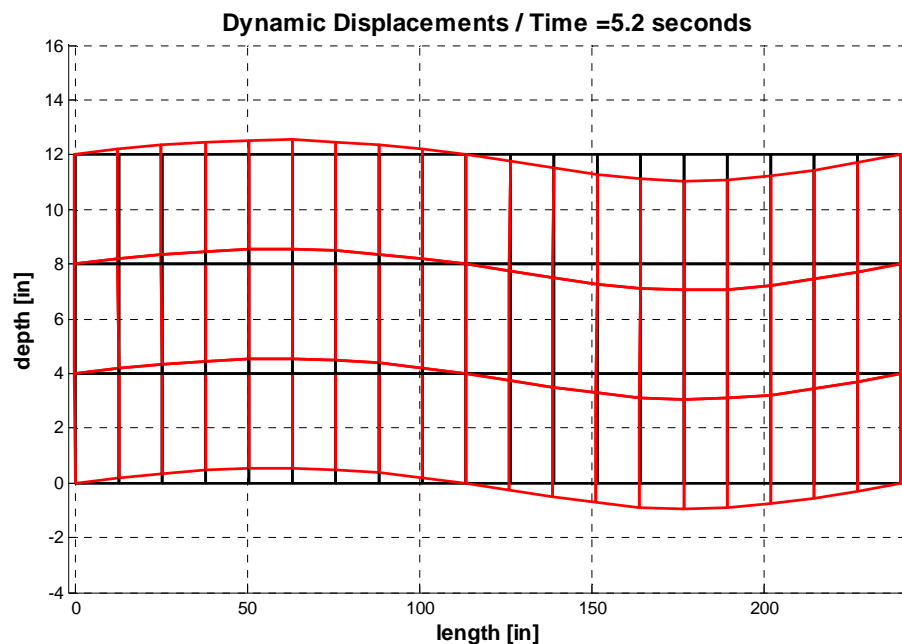


Figure 4-12(d): Case Study 3 – Displacements

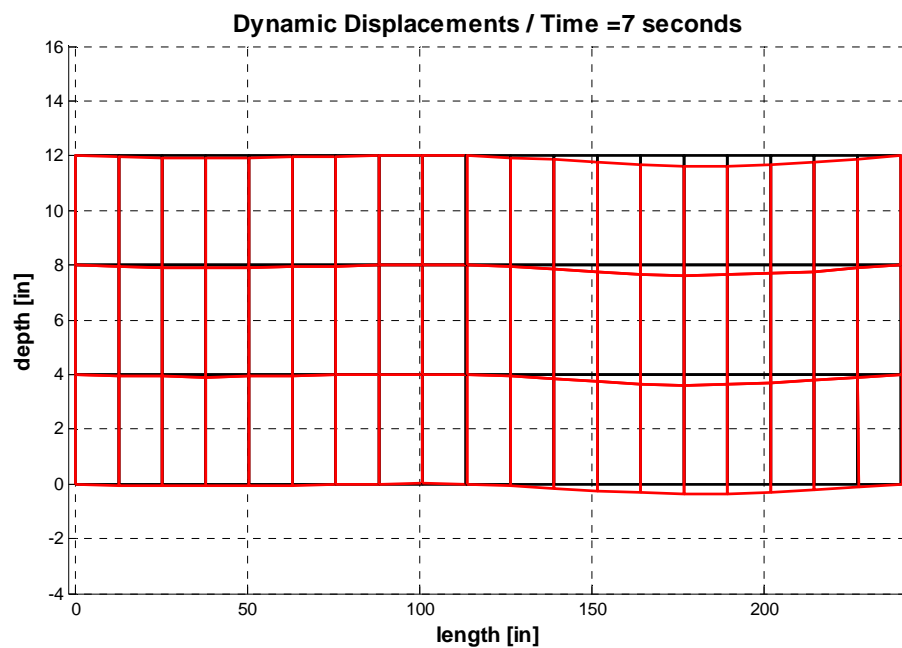


Figure 4-12(e): Case Study 3 – Displacements

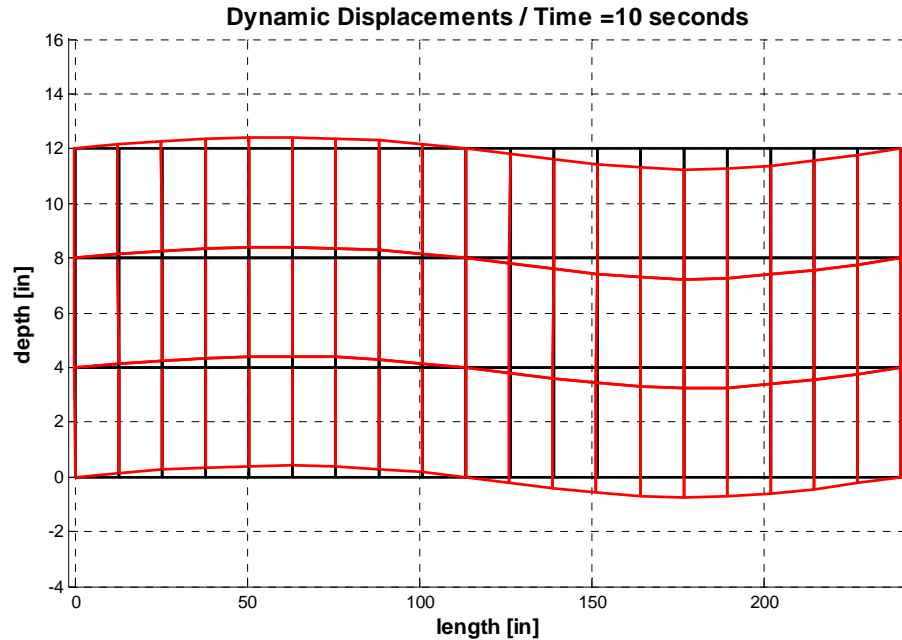


Figure 4-12(f): Case Study 3 – Displacements

Because of the step load, the beam displacement never becomes a positive displacement (with respect to the coordinate system used) between the roller supports and never becomes a negative displacement between the pin and roller supports. These results are as expected.

4.5 Multiple Dynamic Loads

The final case study considers the same beam that was used for the third case study when the step loading was considered. However, the loading condition is different in this case. F_1 represents a harmonic load and F_2 represents a step load. The harmonic load is located 5ft away from the pin support and the step load is located 5ft away from the left end of the beam as shown in the Figure 4-13.

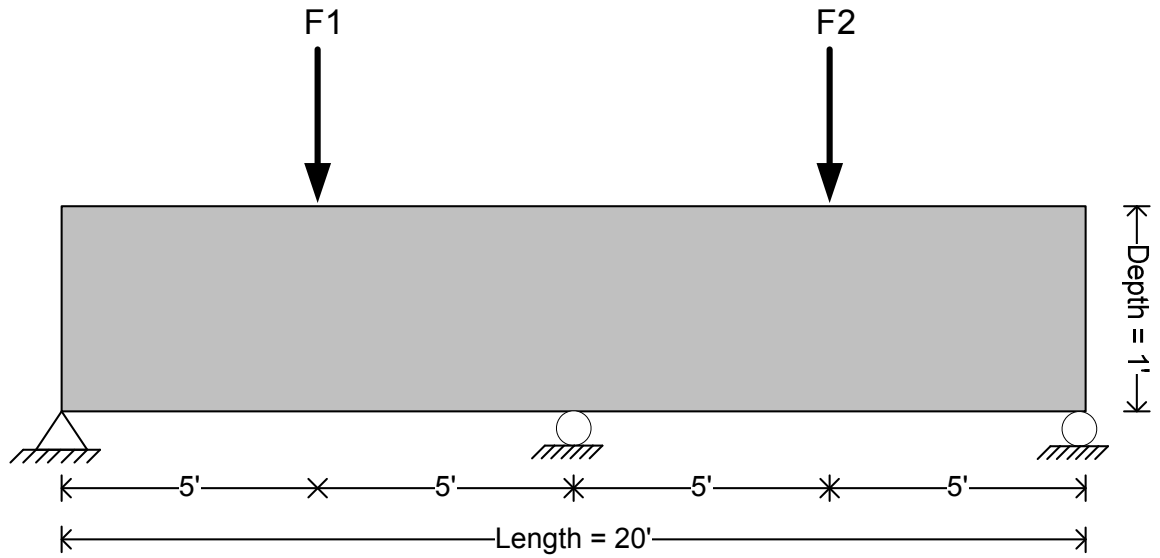


Figure 4-13: Case Study 4 – Continuous Beam

The harmonic load is applied to the beam from time = 0 seconds until 10 seconds and its magnitude varies between -15000 lbs to 15000 lbs. At time = 4 seconds, suddenly another load is applied, the step load, and it stays constant for the duration of time with a magnitude of -20000 lbs. These two time-dependent loads are shown the Figure 4-14.

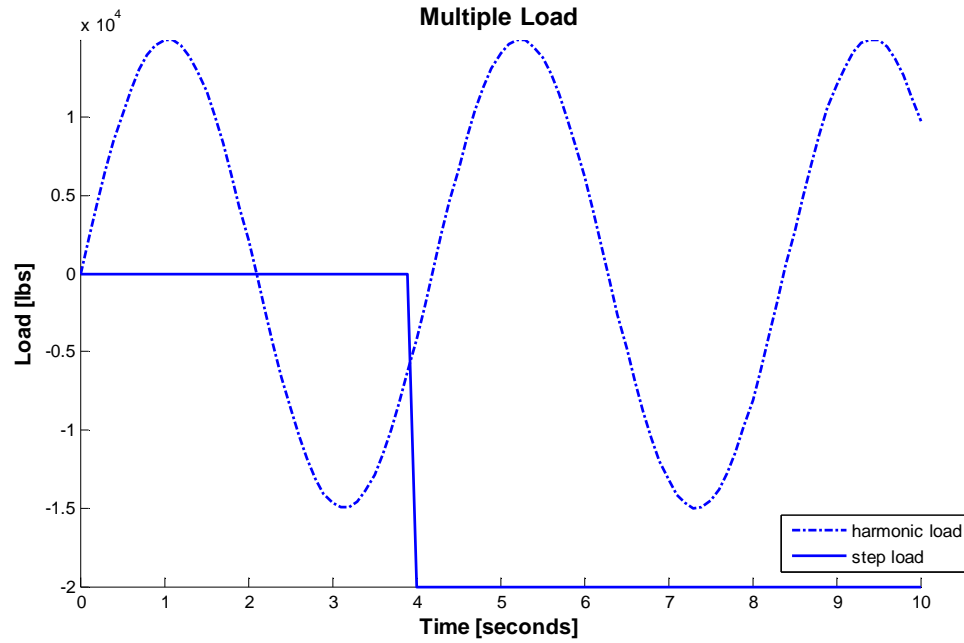


Figure 4-14: Case Study 4 – Multiple Loading

As stated in Table 4-1, the beam is comprised of 57 rectangular Q4 elements and having 160 DOFs. The finite element mesh of the beam is illustrated in the Figure 4-15. Figure 4-16 illustrates the displacement of the beam considering a mesh of Q4 finite elements. Sub-figures (a) – (f) provide a representation of the movie generated at discrete time intervals as indicated on each sub-figure.

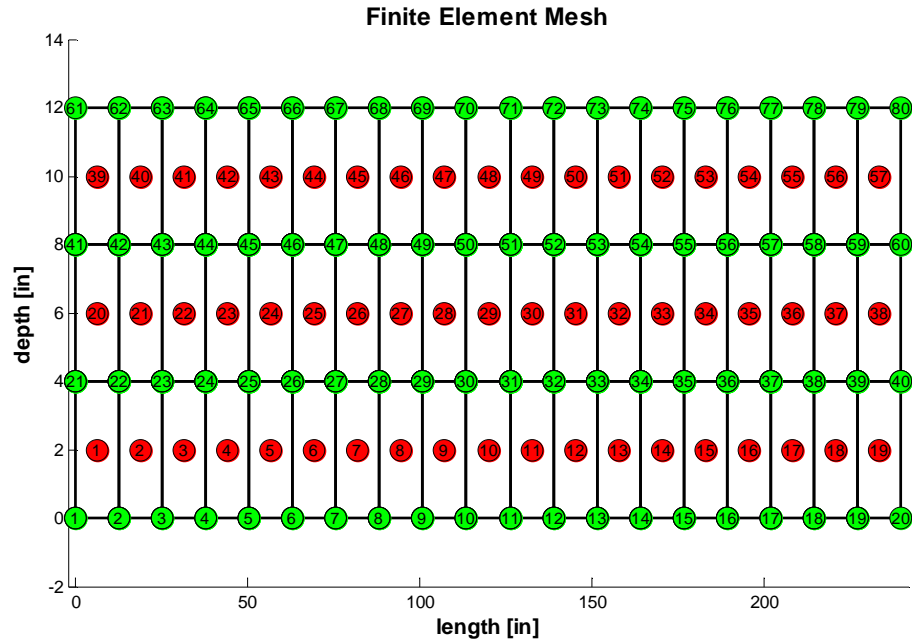


Figure 4-15: Case Study 4 – Finite Element Mesh

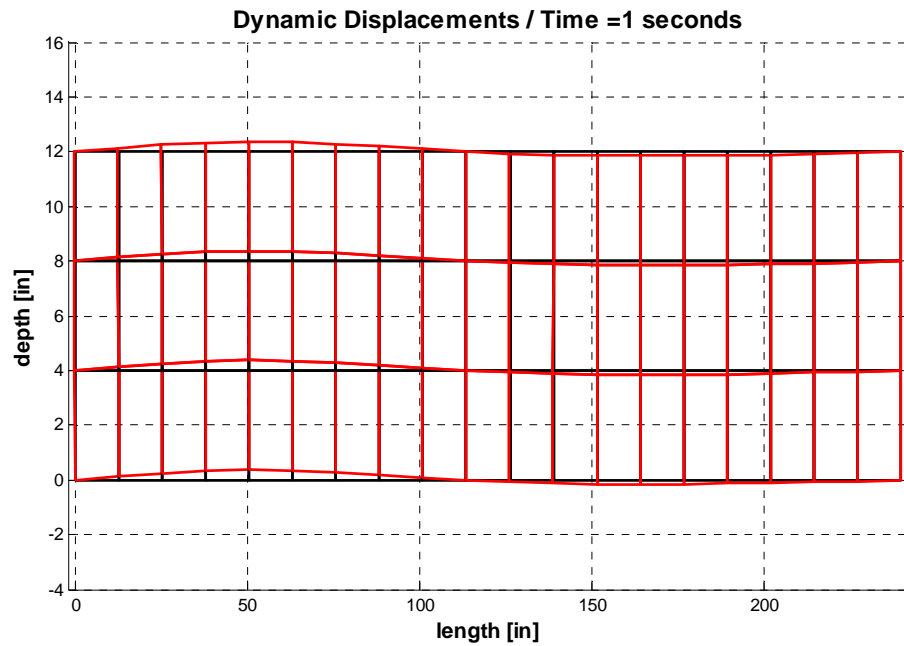


Figure 4-16(a): Case Study 4 – Displacements

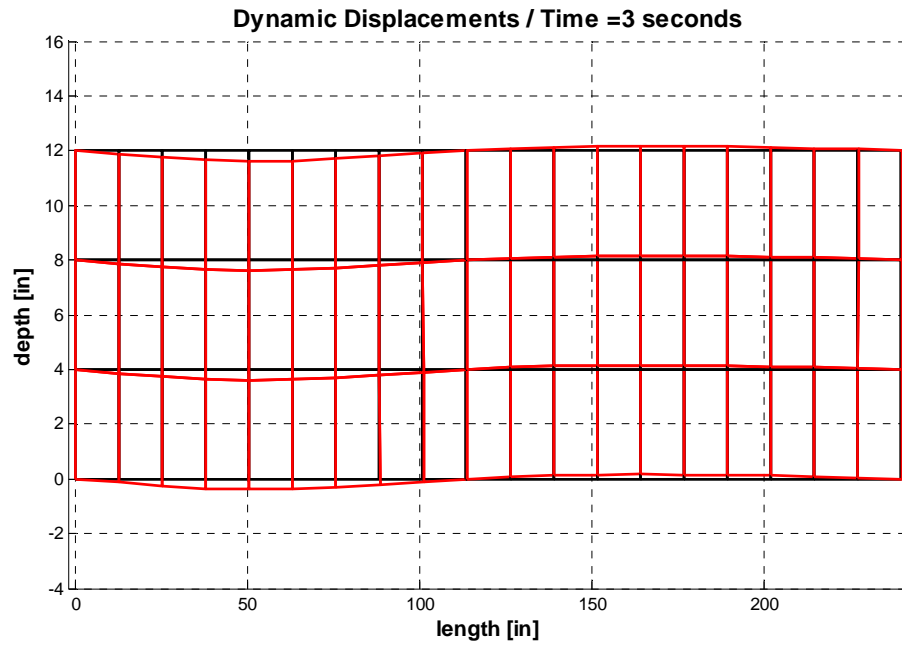


Figure 4-16(b): Case Study 4 – Displacements

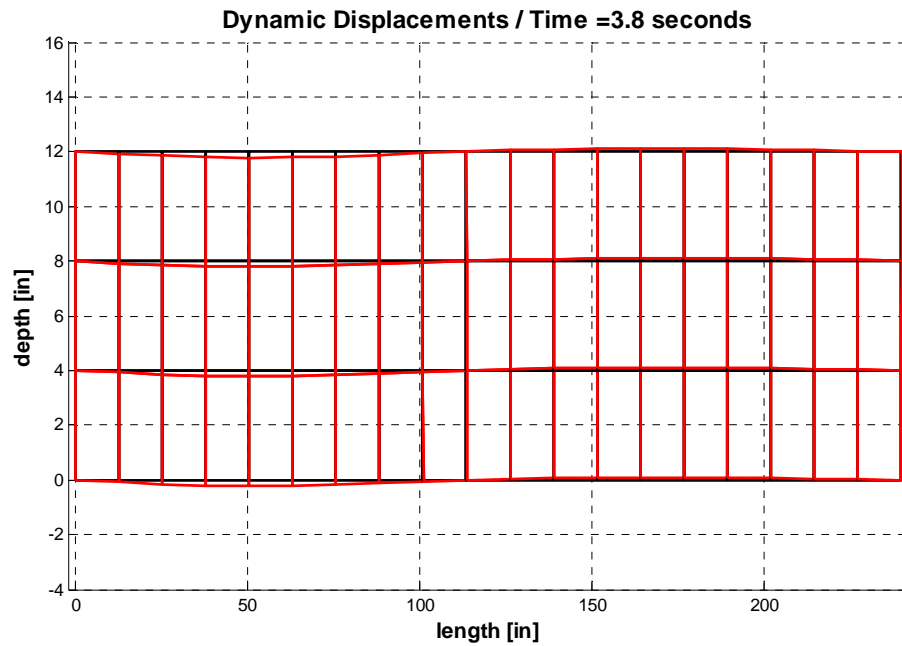


Figure 4-16(c): Case Study 4 – Displacements

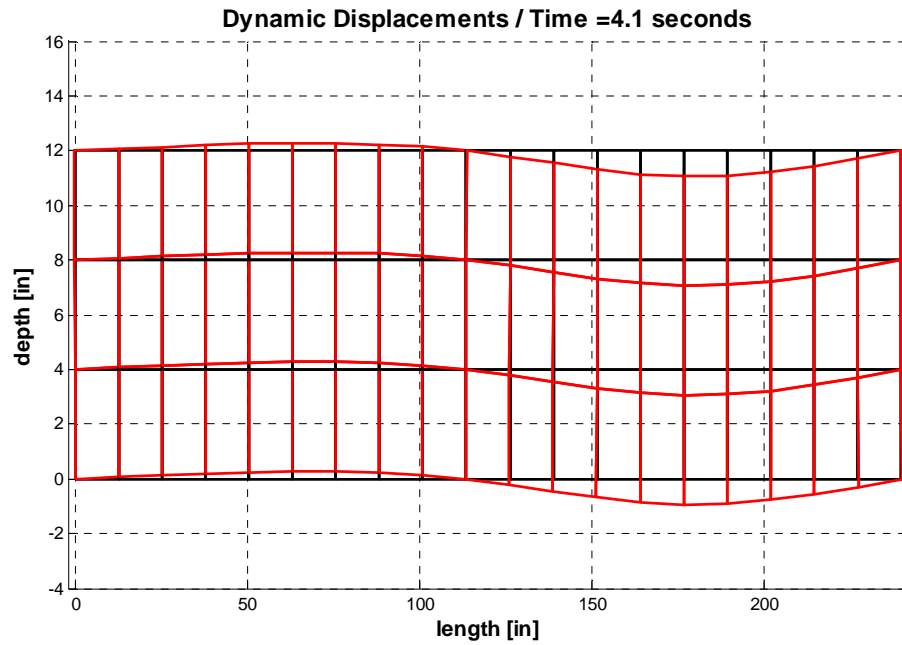


Figure 4-16(d): Case Study 4 – Displacements

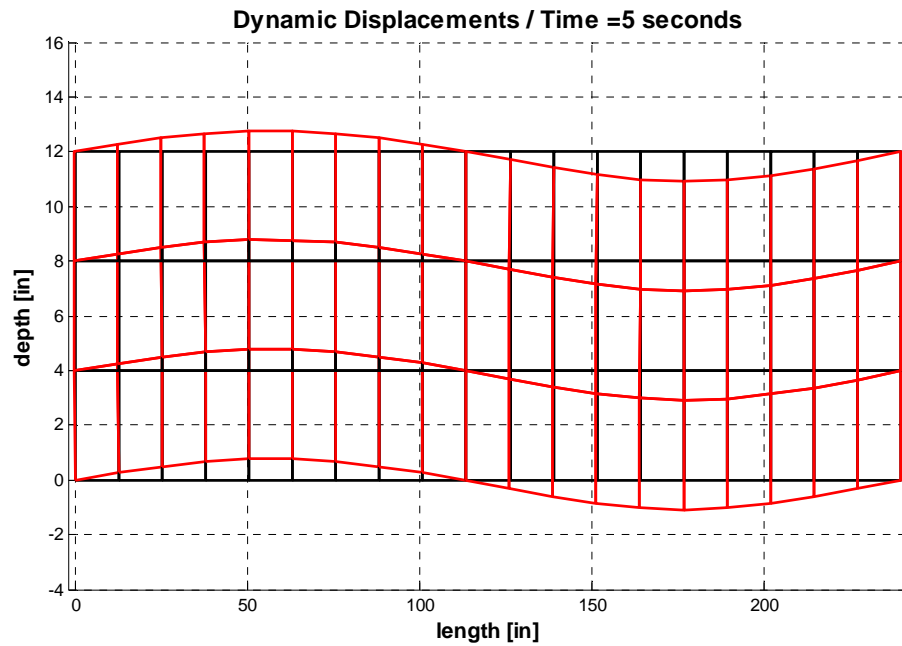


Figure 4-16(e): Case Study 4 – Displacements

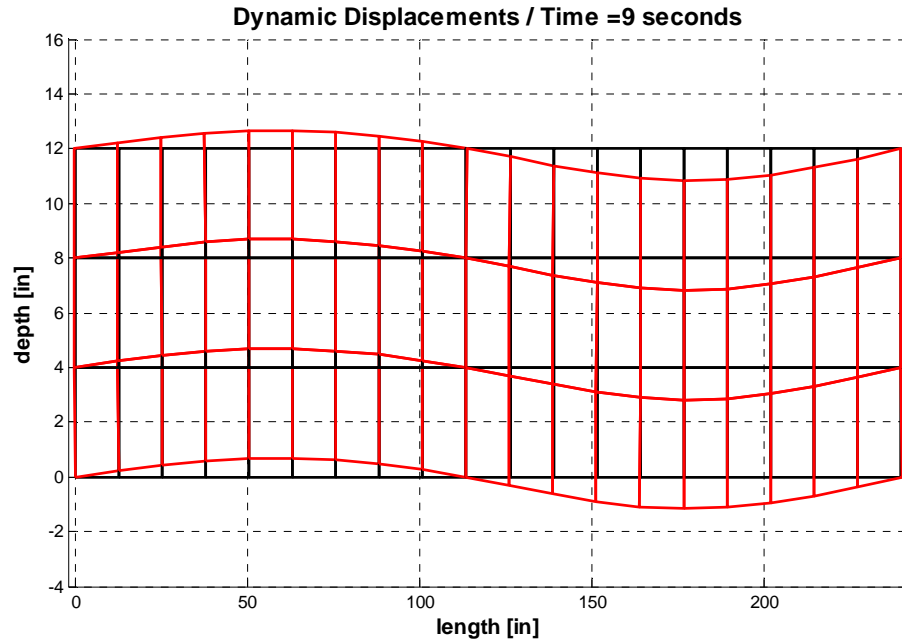


Figure 4-16(f): Case Study 4 – Displacements

As shown in Figure 4-16, the influence of the step load when applied at time = 4 seconds is readily distinguishable. Before the application of the step load, displacements are limited due to the harmonic load. However, clearly the displacement of the beam before 4 seconds is harmonic in nature as illustrated. After a period of time after 4 seconds, the displacement again begins to resemble a harmonic pattern as expected.

CHAPTER V

CONCLUSION

5.1 Summary and Conclusions

The primary objective of this study was to develop a MATLAB based computer program coupling the finite element method and structural dynamics analysis techniques. The primary reason for using the finite element method is its significant advantages in engineering problems. It is a powerful solution technique for differential and integral equations in complex engineering problems. When combined with a structural dynamic solution algorithm, the capabilities of the two become very unique.

Theoretical equations are derived for both the finite element method and structural dynamics during this study. Fundamentally, formulation of element matrices, integration techniques, boundary conditions, and time-stepping methods are discussed and their equations are presented. These concepts are used to develop the computer code.

Several case studies are analyzed during this study with different types of dynamic loads including harmonic loading, impulse loading, and step loading. These loads are also applied to a variety of different beams including a cantilever beam, a simply-supported

beam and a continuous beam with three supports. Displacements and deformed shapes are found using MATLAB FEA program written as a consequence of this study.

This study presents a discussion of the finite element method for structural dynamics applications. It provides a basic understanding of the behavior of structural systems under different dynamic loads. The computer code, which is written in MATLAB, allows for further flexibility not explicitly discussed in this study such as using different material properties throughout the structure, a variety of geometrical shapes of the structure, varying types of loads, and multiple finite element types beyond the Q4 element used in the finite element mesh.

5.2 Recommendations

This study is open to development and further development is expected in the future. Although currently it is limited to Q4 elements for beams, it can be expanded to advanced finite elements and other types of structures (see Iqbal, 2009). The MATLAB code is written to be expanded to allow for the analysis of more complicated structural systems utilized more advanced finite elements and finite element formulations. Moreover, one can develop this study in not only finding displacements, deformed shapes, stresses, and strains but also other topics such as calculating fracture, durability or even reliability of structures.

Although the computer code has good properties, it is also recommended that better meshing capabilities may be implemented in the MATLAB code. The computer code covers linear programming for now. Therefore, the computer program can be incorporated with nonlinearity. If these modifications can be done, this computer code can become a more powerful finite element analysis program.

Work done in conjunction with this study includes the incorporation of advanced finite element formulations beyond the Q4 element used herein (Iqbal, 2009).

REFERENCES

- [1] Bozdogan, K.B., Ozturk, D., and Nuhoglu, A. (2009), “An Approximate Method for Static and Dynamic Analyses of Multi-bay Coupled Shear Walls”, *The Structural Design of Tall and Special Buildings*, Vol. 18, pp. 1-12.
- [2] Cerioni, R., Brighenti, R., and Donida, G. (1995), “Use of Incompatible Displacement Modes in a Finite Element Model to Analyze the Dynamic Behavior of Unreinforced Masonry Panels”, *Computers and Structures*, Vol. 57, pp. 47 - 57.
- [3] Chopra, K.A. (2007), “*Dynamics of Structures Theory and Applications to Earthquake Engineering*”, Pearson Prentice Hall.
- [4] Cook, R.D., Malkus, D.S., Plesha, M.E. and Witt, R.J. (2002), “*Concepts and Applications of Finite Element Analysis*”, John Wiley & Sons.
- [5] Du, H., Hitchings, D., and Davies, G.A.O. (1992), “A Finite Element Structural Dynamics Model of a Beam with an Arbitrary Moving Base – Part I: Formulations”, *Finite Elements in Analysis and Design*, Vol. 12, pp. 133 - 150.
- [6] Kiral, Z., and Kiral, B.G. (2008). “Dynamic Analysis of a Symmetric Laminated Composite Beam Subjected to a Moving Load with Constant Velocity”, *Journal of Reinforced Plastics and Composites*, Vol. 27, pp. 19-32.

- [7] Li, Q.S., Yang, L.F., and Li, G.Q. (2001), “The Quadratic Finite Element/Strip with Generalized Degrees of Freedom and Their Application”, *Finite Elements in Analysis and Design*, Vol. 37, pp. 325 - 339.
- [8] Mermertas, V. (1998), “Dynamic Interaction Between the Vehicle and Simply Supported Curved Bridge Deck”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 162, pp. 125 - 131.
- [9] Ozcan, D.M., Bayraktar, A., Sahin, A., Haktanir, T., and Turker, T. (2008), “Experimental and Finite Element Analysis on the Steel Fiber-Reinforced Concrete (SFRC) Beams Ultimate Behavior. *Construction and Building Materials*, Vol. 23, pp. 1064 - 1077.
- [10] Iqbal, R. (2009), “*Development of a Finite Element Program Incorporating Advanced Element Types*”, MS Thesis, Oklahoma State University.

APPENDIX A

MATLAB FEA PROGRAM

The Matlab FEA program that was developed during the course of this study is provided herein. First, the main FEA program (not containing the sub-routines since the FEA program is still under development) is provided. Second, the general input file used to setup an analysis is provided. Both of these files are Matlab script files and can be executed in all versions of Matlab. To generate graphical output, the full version of Matlab is required.

Main Program

```
% *****
% Finite Element Analysis (FEA) Program to Determine the Structural and Thermal Response of Structural Systems
% Written By: Muhammet Saglar & Rameez Iqbal (Advised By: Dr. Jonathan S. Goode)
% School of Civil and Environmental Engineering
% Oklahoma State University
% *****

% *****
% *****
% Begin Program

% Clear Variables and Settings
clear all

% Clear Screen
clc;

% Inhibit Warning Messages
warning('off','all');

disp('*****');
disp('Finite Element Analysis');
disp('Written by: Saglar/Iqbal (Goode)');
disp('School of Civil and Environmental Engineering');
disp('Oklahoma State University');
disp('*****');
```

```

disp('*****');
disp('Running Analysis .....');
disp('*****');

% Begin Timer
tic;

% *****
% Pre-Processing Section
% *****

disp('Pre-Processing .....');

% Call M-File for User Input Data from fea_input.m
disp('..... Obtaining Input .....');
fea_input;

% Call M-File for Initialization of Variables from fea_initialize.m
disp('..... Initializing Program .....');
fea_initialize;

% *****
% End Pre-Processing Section
% *****

% *****
% Processing Section
% *****

disp('Processing .....');

% -----
% Assembly Process

disp('..... Assembly Process .....');

% Determine Element Property Matrices (Constitutive, Mass, Conductivity)
[property_matrix_k,property_matrix_m,property_matrix_t] =
fea_property_matrix(analysis_type_material,material_E,material_v,material_rho,material_t);

% Determine Gauss Quadrature Sampling Points and Weights
[gauss_points,gauss_weights] = fea_gauss_2d(element_type);

% Initialize Global Index of DOF
index_global = zeros(edof,nel);

% Assemble System Matrices According to Analysis
if analysis_type == 'S'

    % Static Analysis

    disp('..... Static Analysis .....');
    disp('..... Building Stiffness .....');

    % Loop Through All Elements to Determine System Matrices (Stiffness, Mass, and Conductivity)
    % NOTE: Depending on the Analysis Type, Some Matrices Will Not be Generated
    for j = 1:nel

        % Call Sub-Routine to Extract Nodes for the j-th Element
        [element_nodes,element_xcoord,element_ycoord] = fea_node_coordinates(j,nel,node_connectivity,node_coord);

        % Call Sub-Routine to Extract System DOFs for the j-th Element
        index = fea_index_dof(element_nodes,nel,ndof,edof);

        % Save Index of DOF to Global Variable
        index_global(:,j) = index;

        % Call Sub-Routine to Generate Element Stiffness Matrix

```

```

        element_k =
fea_element_matrix_stiffness(element_type,gauss_points,gauss_weights,element_xcoord,element_ycoord,nnel,edof,property_matrix_
k);

        % Call Sub-Routine to Assemble System Stiffness Matrix
        system_k = fea_element_matrix_assemble(system_k,element_k,index,edof);

    end

elseif analysis_type == 'D'

    % Dynamic Analysis

    disp('..... Dynamic Analysis .....');
    disp('..... Building Stiffness and Mass .....');

    % Loop Through All Elements to Determine System Matrices (Stiffness, Mass, and Conductivity)
    % NOTE: Depending on the Analysis Type, Some Matrices Will Not be Generated
    for j = 1:nnel

        % Call Sub-Routine to Extract Nodes for the j-th Element
        [element_nodes,element_xcoord,element_ycoord] = fea_node_coordinates(j,nnel,node_connectivity,node_coord);

        % Call Sub-Routine to Extract System DOFs for the j-th Element
        index = fea_index_dof(element_nodes,nnel,ndof,edof);

        % Save Index of DOF to Global Variable
        index_global(:,j) = index;

        % Call Sub-Routine to Generate Element Stiffness Matrix
        element_k =
fea_element_matrix_stiffness(element_type,gauss_points,gauss_weights,element_xcoord,element_ycoord,nnel,edof,property_matrix_
k);

        % Call Sub-Routine to Assemble System Stiffness Matrix
        system_k = fea_element_matrix_assemble(system_k,element_k,index,edof);

        % Call Sub-Routine to Generate Element Mass Matrix
        element_m =
fea_element_matrix_mass(element_type,gauss_points,gauss_weights,element_xcoord,element_ycoord,nnel,edof,property_matrix_m);

        % Call Sub-Routine to Assemble System Mass Matrix
        system_m = fea_element_matrix_assemble(system_m,element_m,index,edof);

    end

end

% End Assembly Process
% -----

% -----
% Boundary and Loading Conditions

disp('..... Applying Boundary and Loading Conditions .....');

% Apply Boundary and Loading Conditions According to Analysis
if analysis_type == 'S'

    % Static Analysis

    % Call Sub-Routine to Apply Boundary and Loading Conditions
    [system_k,system_f] = fea_system_boundary_load(system_k,system_f,boundary_dof,boundary_dof_val,load_dof,load_dof_val);

elseif analysis_type == 'D'

    % Dynamic Analysis

    % Boundary and Loading Conditions Must Be Applied at Each Point in Time During the Dynamic Analysis
    % See the Analysis Section Below

```

```

end

% End Boundary Conditions
% -----

% -----
% Analysis

disp('..... Solving .....');

% Based on Analysis Type
if analysis_type == 'S'

    % Static Analysis

    disp('..... Static Analysis .....');

    % Solve System Equations for Static Displacements
    system_d = system_k \ system_f;

elseif analysis_type == 'D'

    % Dynamic Analysis

    disp('..... Dynamic Analysis .....');

    % Call Sub-Routine to Solve System Equations for Dynamic Displacements
    [system_d,system_d_vel,system_d_acc,system_f,dynamic_time] =
fea_dynamic_analysis(system_k,system_m,boundary_dof,boundary_dof_val,load_dof,load_file,dynamic_delta_t,dynamic_beta,dyna
mic_gamma,sdof);

end

% End Analysis
% -----

% *****
% End Processing Section
% *****

% *****
% Post-Processing Section
% *****

disp('Post-Processing .....');

% -----
% Plot Generation

% Plot Finite Element Mesh
if plot_element_mesh == 'Y'

    disp('..... Plotting Element Mesh .....');

    % Call Sub-Routine to Plot Finite Element Mesh
    plot_handle =
fea_plot_mesh(proj_name,nel,node_coord,node_connectivity,plot_element_mesh_numbers,plot_element_mesh_node_numbers);

end

% Based on Analysis Type
if analysis_type == 'S'

    % Static Analysis

    disp('..... Static Analysis .....');

    % Plot Static Displacements
    if plot_static_displacement == 'Y'

```

```

disp('..... Plotting Static Displacements .....');

% Call Sub-Routine to Plot Static Displacements
plot_handle = fea_plot_static_displacement(proj_name,nel,node_coord,node_connectivity,system_d,index_global);

if plot_static_contour_displacement == 'Y'

    % Call Sub-Routine to Plot Displacement Contours
    plot_handle = fea_plot_static_displacement_contour(proj_name,node_coord,node_connectivity,system_d,sdof,nel);

end

end

% Determine Static Stresses and Strains for Plots
if plot_static_stress == 'Y' || plot_static_strain == 'Y'

    disp('..... Determining Static Stresses and Strains .....');

    % Call Sub-Routine to Determine Stresses and Strains from Displacements
    [system_stress,system_strain,gauss_points_coord] =
fea_stress_strain(nel,nnel,edof,node_connectivity,node_coord,index_global,system_d,property_matrix_k,element_type,gauss_points);

end

% Plot Static Stresses
if plot_static_stress == 'Y'

    disp('..... Plotting Static Stresses .....');

    % Call Sub-Routine to Plot Static Stresses
    plot_handle = fea_plot_static_stress(proj_name,nel,node_coord,node_connectivity,system_stress,gauss_points_coord);

    if plot_static_contour_stress

        % Call Sub-Routine to Plot Stress Contours
        [plot_handle] =
fea_plot_static_stress_contour(proj_name,nel,node_coord,node_connectivity,system_stress,gauss_points_coord);

    end

end

% Plot Static Strains
if plot_static_strain == 'Y'

    disp('..... Plotting Static Strains .....');

    % Call Sub-Routine to Plot Static Strains
    plot_handle = fea_plot_static_strain(proj_name,nel,node_coord,node_connectivity,system_strain,gauss_points_coord);

    if plot_static_contour_strain

        % Call Sub-Routine to Plot Strain Contours
        [plot_handle] =
fea_plot_static_strain_contour(proj_name,nel,node_coord,node_connectivity,system_strain,gauss_points_coord);

    end

end

elseif analysis_type == 'D'

    % Dynamic Analysis

    disp('..... Dynamic Analysis .....');

    % Call Sub-Routine to Plot Dynamic Displacements (Movie)

```

```

        plot_handle =
fea_plot_dynamic_displacement(proj_name,nel,node_coord,node_connectivity,system_d,dynamic_time,index_global);

        disp('..... Plotting Dynamic Displacements .....');

    end

% *****
% End Post-Processing Section
% *****

disp('*****');
disp('Analysis Done .....');
disp('*****');

% End Timer
analysis_time = toc;

disp(['Total Time of Analysis: ',num2str(analysis_time),' seconds']);
disp('*****');

% End Program
% *****
% *****

```

Input File

```

% *****
% Finite Element Analysis (FEA) - Input File
% Written By: Muhammet Saglar & Rameez Iqbal (Advised By: Dr. Jonathan S. Goode)
% School of Civil and Environmental Engineering
% Oklahoma State University
% *****

% *****
% *****
% Begin Input File

% -----
% Project Name

% Define Project Name
% NOTE: Automatically Saved Figures and Data Will Be Proceeded with the Project Name
proj_name = 'test';

% -----
% Analysis Options

% Define Analysis Type
% Analysis Type Options:
% Input 'S' = Static Analysis
% Input 'D' = Dynamic Analysis
% Input 'T' = Thermal Analysis
analysis_type = 'S';

% -----
% Input Finite Element Mesh Properties

% Input Element Type
% Input 'Q4' for Bilinear Rectangular Element
% Input 'Q8' for Quadratic Rectangular Element
% Input 'CST' for Constant Strain / Linear Triangular Element
% Input 'LST' for Linear Strain / Quadratic Triangular Element

```

```

element_type = 'Q4';

% Input Node Coordinates (x and y coordinates) [Note Units => Length]
node_coord = [ 0.0  0.0  ;...
               12.0 0.0  ;...
               24.0 0.0  ;...
               36.0 0.0  ;...
               48.0 0.0  ;...
               60.0 0.0  ;...
               72.0 0.0  ;...
               84.0 0.0  ;...
               96.0 0.0  ;...
               108.0 0.0 ;...
               120.0 0.0 ;...
               0.0 12.0 ;...
               12.0 12.0 ;...
               24.0 12.0 ;...
               36.0 12.0 ;...
               48.0 12.0 ;...
               60.0 12.0 ;...
               72.0 12.0 ;...
               84.0 12.0 ;...
               96.0 12.0 ;...
               108.0 12.0 ;...
               120.0 12.0 ];

% Input Nodal Connectivity for Each Element (CCW from Bottom-Left)
node_connectivity = [ 1  2  13 12  ;...
                     2  3  14 13  ;...
                     3  4  15 14  ;...
                     4  5  16 15  ;...
                     5  6  17 16  ;...
                     6  7  18 17  ;...
                     7  8  19 18  ;...
                     8  9  20 19  ;...
                     9  10 21 20  ;...
                     10 11 22 21  ];

% -----
% Input Boundary Conditions

% Input DOF Constrained
boundary_dof = [1 2 23 24];

% Input Constrained DOF Prescribed Values [Note Units => Length]
% NOTE: For Dynamic Analysis, All Prescribed Values MUST BE Zero (0)
boundary_dof_val = [0 0 0 0];

% -----
% Input Material Properties

% Material Properties
% NOTE: If the property is not being used, enter 0 (zero)

% Modulus of Elasticity [Note Units => Force / Length^2]
material_E = 1e6;

% Poisson's Ratio [Unitless]
material_v = 0.3;

% Mass Density [Note Units => Mass / Length^3]
material_rho = 1;

% Define Material Analysis Type
% Material Analysis Type Options:
% Input '1' = Plane Stress Analysis (2-D)
% Input '2' = Plane Strain Analysis (2-D)
% Input '3' = Three-Dimensional Analysis (3-D)
analysis_type_material = 1;

```



```

% -----
% Input Load Properties

% Static Loading (If Applicable)
% NOTE: If Static Analysis is not being considered, NO changes are needed

if strcmp(analysis_type,'S') == 1

    % Input DOF Statically Loaded
    load_dof = [18 20];

    % Input DOF Statically Loaded Prescribed Values [Note Units => Force]
    load_dof_val = [-500 -500];

% Dynamic Loading (If Applicable)
% NOTE: If Dynamic Analysis is not being considered, NO changes are needed

elseif strcmp(analysis_type,'D') == 1
    % Input DOF Dynamically Loaded
    load_dof = [44];

    % Input Load Time-Series File Name [Note Units => Force]
    % NOTE: First Number Designates the Load at Time = 0 seconds
    % Column 1 -> n = DOFs
    % Row 1 -> n = Load at Time Increments
    load_file = 'load.dat';

    % Input Time Step of Load Time-Series [seconds]
    dynamic_delta_t = 0.1;

    % Define Newmark-Beta Parameters [Unitless]
    % Average Acceleration Method => Beta = 0.25 & Gamma = 0.5 (Unconditionally Stable)
    % Linear Acceleration Method => Beta = 0.16667 & Gamma = 0.5 (Conditionally Stable -> delta_t / Tn <= 0.551)
    dynamic_beta = 0.25;
    dynamic_gamma = 0.5;

end

% -----
% Output Options

% General Plots

% Plot Element Mesh (Y or N)
plot_element_mesh = 'Y';

% Plot Element Mesh Options

% Display Element Numbers on Element Mesh Plot (Y or N)
plot_element_mesh_numbers = 'Y';

% Display Element Node Numbers on Element Mesh Plot (Y or N)
plot_element_mesh_node_numbers = 'Y';

% Static Analysis (If Applicable)
% NOTE: If Static Analysis is not being considered, NO changes are needed

% Plot Static Displacements (Y or N)
plot_static_displacement = 'Y';

% Contour Plots of Displacements (Y or N)
% NOTE: Must Plot Static Displacements to Plot Contours
plot_static_contour_displacement = 'Y';

% Plot Static Stresses (Y or N)
plot_static_stress = 'N';

% Contour Plots of Stresses (Y or N)
% NOTE: Must Plot Static Stresses to Plot Contours
plot_static_contour_stress = 'Y';

```

```

% Plot Static Strains (Y or N)
plot_static_strain = 'N';

% Contour Plots of Strains (Y or N)
% NOTE: Must Plot Static Strains to Plot Contours
plot_static_contour_strain = 'Y';

% Dynamic Analysis (If Applicable)
% NOTE: If Dynamic Analysis is not being considered, NO changes are needed

% Movie of Dynamic Displacements (Y or N)
movie_dynamic_displacement = 'Y';

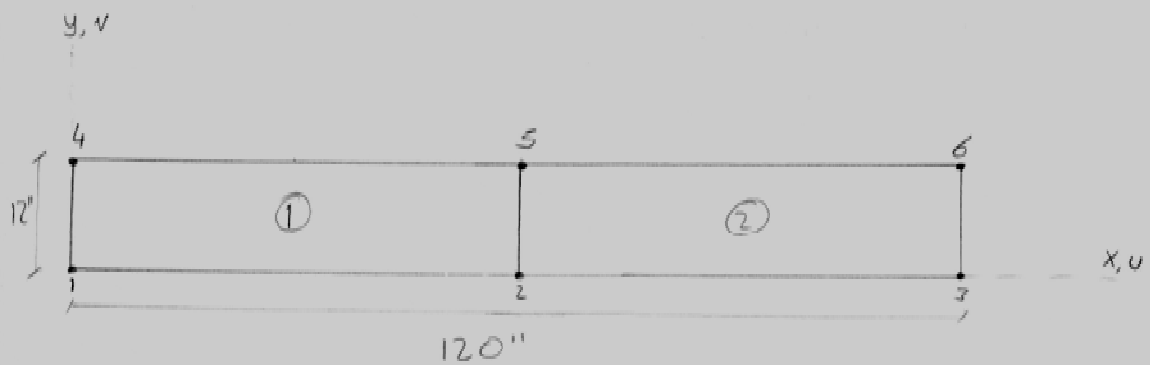
% End Input File
% *****
% *****

```

APPENDIX B

CONFIRMATION CALCULATIONS

Appendix B provides confirmation calculations of the element formulation for the Q4 elements as well as a few select time intervals for the structural dynamics response of the structural system. These calculations were done by hand and then compared to the results developed by the Matlab FEA program as provided in Appendix A.



6 nodes , 12 degrees of freedom, 2 Q4 element.

Modulus of Elasticity = $E = 1e6$ pound / in²

Poisson Ratio = $\nu = 0.3$, Mass Density (ρ) = 1 pound / in

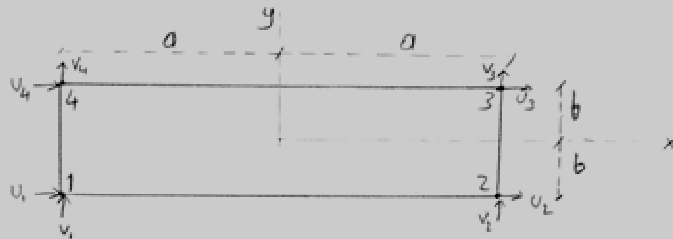
Thickness is assumed to be 1" (2D analysis) .

Length of beam = 120"

Depth of beam = 12"

3 problems are presented. First two are static analysis and the third problem is dynamic analysis.

For a Q4 element (local element):



$$a = 30'' \quad , \quad b = 6''$$

Shape functions are ;

$$N_1 = \frac{(a-x)(b-y)}{4ab} \quad , \quad N_2 = \frac{(a+x)(b-y)}{4ab}$$

$$N_3 = \frac{(a+x)(b+y)}{4ab} \quad , \quad N_4 = \frac{(a-x)(b+y)}{4ab}$$

$$[k]_{8 \times 8} = \int_{-b}^b \int_{-a}^a [B]^T [E] [B] t \, dx \, dy$$

(element stiffness matrix)

where ;

$$[E] = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix}$$

(Constitutive matrix)

$$[B] = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix}$$

strain-displacement matrix

Substituting the shape functions into the this equation;

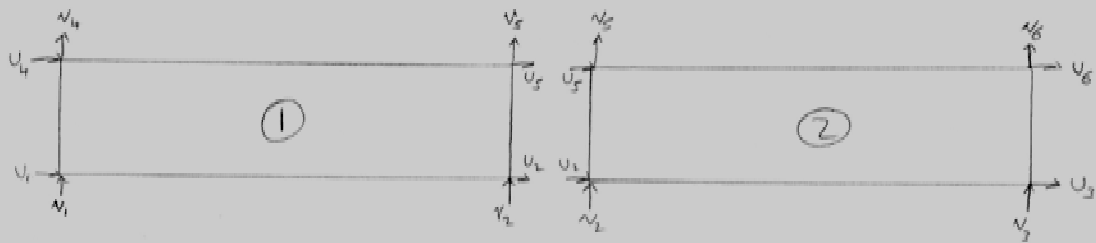
$$\Rightarrow [B] = \frac{1}{4ab} \begin{bmatrix} -(b-y) & 0 & (b-y) & 0 & (b+y) & 0 & -(b+y) & 0 \\ 0 & -(a-x) & 0 & -(a+x) & 0 & (a+x) & 0 & (a-x) \\ -(a-x) & -(b-y) & -(a+x) & (b-y) & (a+x) & (b+y) & (a-x) & -(b+y) \end{bmatrix}$$

substituting them into $[k] = \int_{-b}^b \int_{-a}^a [B]^T [E] [B] t \, dx \, dy$

and after taking integration, the local stiffness matrix

	u1	v1	u2	v2	u3	v3	u4	v4
u1	714285.7	178571.4	247252.7	-13736.3	-357143	-178571	-604396	13736.26
v1	178571.4	1857143	13736.26	890109.9	-178571	-928571	-13736.3	-1818681
u2	247252.7	13736.26	714285.7	-178571	-604396	-13736.3	-357143	178571.4
v2	-13736.3	890109.9	-178571	1857143	13736.26	-1818681	178571.4	-928571
u3	-357143	-178571	-604396	13736.26	714285.7	178571.4	247252.7	-13736.3
v3	-178571	-928571	-13736.3	-1818681	178571.4	1857143	13736.26	890109.9
u4	-604396	-13736.3	-357143	178571.4	247252.7	13736.26	714285.7	-178571
v4	13736.26	-1818681	178571.4	-928571	-13736.3	890109.9	-178571	1857143

$k =$



The stiffness matrices are same for both elements.

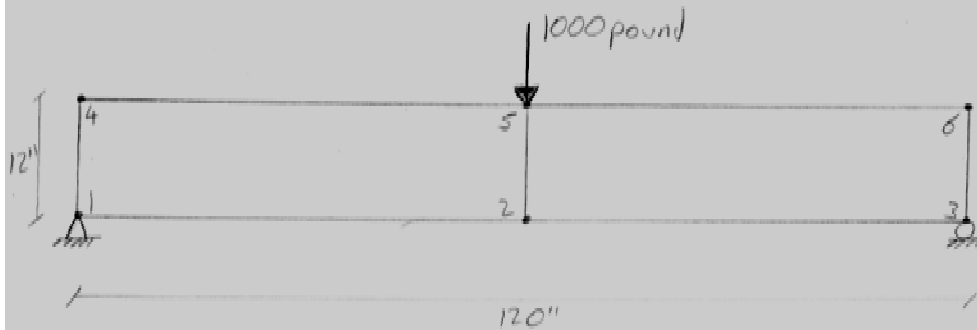
⇒ After mapping these two element's stiffness matrices, we can find global stiffness matrix

	u1	v1	u2	v2	u3	v3	u4	v4	u5	v5	u6	v6
u1	714285.7	178571.429	247252.747	-13736.264	0	0	-604395.6	13736.26	-357142.86	-178571.43	0	0
v1	178571.4	1857142.86	13736.2637	890109.89	0	0	-13736.26	1818681.3	178571.43	928571.43	0	0
u2	247252.7	13736.2637	1428571.43	-2.91E-11	247252.7	-13736.264	-357142.9	178571.4	-1208791.2	0	-357142.9	-178571.43
v2	-13736.26	890109.89	-2.91E-11	3714285.71	13736.26	890109.89	178571.4	-928571.4	0	-3637362.6	-178571.4	-928571.43
u3	0	0	247252.747	13736.2637	714285.7	-178571.43	0	0	-357142.86	178571.429	-604395.6	-13736.264
v3	0	0	-13736.264	890109.89	-178571.4	1857142.86	0	0	178571.429	-928571.43	13736.26	-1818681.3
u4	-604395.6	-13736.264	-357142.86	178571.429	0	0	714285.7	-178571.4	247252.747	13736.2637	0	0
v4	13736.26	-1818681.3	178571.429	-928571.43	0	0	-178571.4	1857143	-13736.264	890109.89	0	0
u5	-357142.9	-178571.43	-1208791.2	3.64E-12	-357142.9	178571.429	247252.7	-13736.26	1428571.43	2.91E-11	247252.7	13736.2637
v5	-178571.4	-928571.43	-1.82E-12	-3637362.6	178571.4	-928571.43	13736.26	890109.9	2.91E-11	3714285.71	-13736.26	890109.89
u6	0	0	-357142.86	-178571.43	-604395.6	13736.2637	0	0	247252.747	-13736.264	714285.7	178571.429
v6	0	0	-178571.43	-928571.43	-13736.26	-1818681.3	0	0	13736.2637	890109.89	178571.4	1857142.86

Static Analysis Calculations:

Case 1:

- Simply supported beam and loaded at the midpoint:



⇒ Applying boundary condition: At the first node, vertical and horizontally, at the third node only vertically, we should make them too stiff. To do that we may add $1e30$ to U_1 , V_1 and V_3 in the system stiffness matrix.

⇒ Force matrix: We may apply the load at the 5th and 2nd nodes equally (500 pound for each).

To find displacement ;

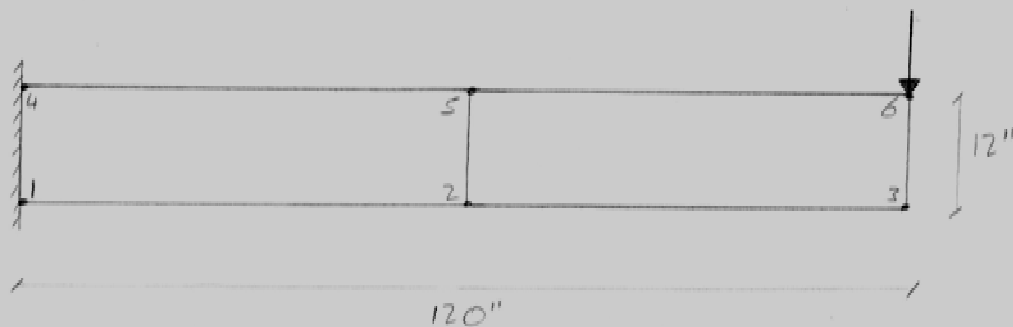
$$[K]\{D\} = \{R\} \Rightarrow \text{we already know stiffness matrix } [K] \text{ and force matrix } \{R\}$$

\Rightarrow We can apply gauss elimination to find displacement matrix $\{D\}$.

$$\{R\} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ -500 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -500 \\ 0 \\ 0 \end{Bmatrix} \text{ (pounds)} \Rightarrow \{D\} = \begin{Bmatrix} 0 \\ 0 \\ 0.0036 \\ -0.0241 \\ 0.0071 \\ 0 \\ 0.007 \\ 0.00018 \\ 0.0036 \\ -0.024 \\ 0.00015 \\ -0.00018 \end{Bmatrix} \text{ (inches)}$$

Case 2:

- Fixed beam at the left end and loaded at the right end.



⇒ Applying boundary condition: At the 1st and the 4th nodes, vertically and horizontally we should make them too stiff. So, we may add $1e30$ to u_1, v_1, u_4, v_4 in the system stiffness matrix.

⇒ Force matrix: We may apply the load at the 6th and 3rd nodes equally (500 pound for each).

$$\Rightarrow \{R\} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -500 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -500 \end{Bmatrix} \quad (\text{pounds})$$

To find the displacement ;

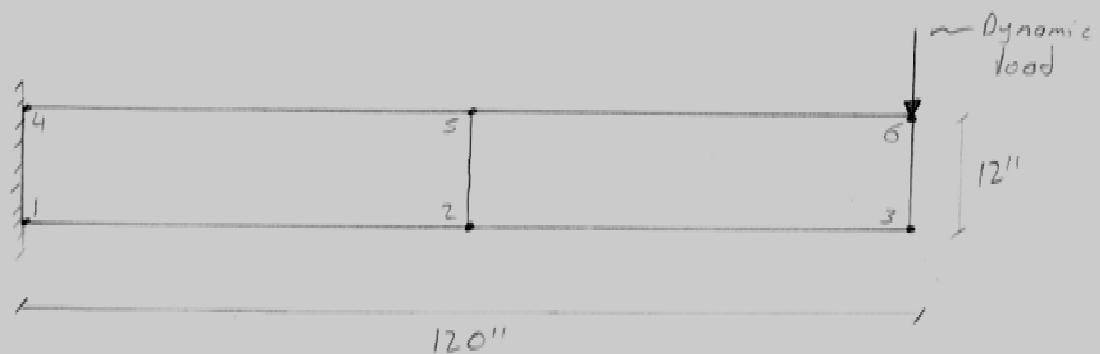
$$[K]\{D\} = \{R\} \Rightarrow \text{some procedure with case 1.}$$

$$\Rightarrow \{D\} = \begin{Bmatrix} 0 \\ 0 \\ -0.021 \\ -0.118 \\ -0.028 \\ -0.376 \\ 0 \\ 0 \\ 0.021 \\ -0.118 \\ 0.028 \\ -0.376 \end{Bmatrix} \quad (\text{inches})$$

Comparison of Hand Calculation and Computer Results - Case 1		
DOF	Computer Results (in)	Hand Calculation Results (in)
u1	-7.32E-48	0
u2	-2.69E-34	0
u3	0.003575	0.0036
u4	-0.024134615	-0.0241
u5	0.00715	0.0071
u6	-2.69E-34	0
u7	0.007	0.007
u8	-0.000184615	-0.00018
u9	0.003575	0.0036
u10	-0.02405	-0.024
u11	0.00015	0.00015
u12	-0.000184615	-0.00018

Comparison of the Hand Calculation and Computer Results - Case 2		
DOF	Computer Results (in)	Hand Calculation Results (in)
u1	-1.40E-32	0
u2	-2.69E-34	0
u3	-0.021	-0.021
u4	-0.118	-0.118
u5	-0.028	-0.028
u6	-0.376	-0.376
u7	1.40E-32	0
u8	-2.69E-34	0
u9	0.021	0.021
u10	-0.118	-0.118
u11	0.028	0.028
u12	-0.376	-0.376

Dynamic Analysis Calculations:



- Fixed beam from the left end and loaded dynamically at the right end. 2 $\Phi 4$ elements are used.
- Undamped system, $\Delta t = 0.1$ second, final time = 0.4 s
Loads $\Rightarrow 0, -644, -985, -863, -335$ pounds are applied respectively. (sinusoidal force)
- Newmark-Beta method is used.

Newmark parameters $\Rightarrow \gamma = 1/2, \beta = 0.25$

$U \Rightarrow$ displacement, $\dot{U} \Rightarrow$ velocity, $\ddot{U} \Rightarrow$ acceleration

$P \Rightarrow$ force

Calculation of element mass matrix :

$$\underset{\substack{8 \times 8 \\ \text{element mass} \\ \text{matrix}}}{[m]} = \int_{-b}^b \int_{-a}^a \underset{8 \times 2}{[N]}^T \underset{1 \times 1}{[\rho]} \underset{2 \times 8}{[N]} t \, dx \, dy$$

where ; N is shape function matrix and ρ is mass density. Shape functions were mentioned while calculation of element stiffness matrix.

$a = 30''$, $b = 6''$ (previously found in stiffness calculation)

$$N = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \text{ and } \rho = 1 \text{ pound/in.}$$

substituting them into $[m] = \int_{-b}^b \int_{-a}^a [N]^T [\rho] [N] t \, dx \, dy$ and after taking the integration, the local mass matrix is;

$m =$

80	0	40	0	20	0	40	0
0	80	0	40	0	20	0	40
40	0	80	0	40	0	20	0
0	40	0	80	0	40	0	20
20	0	40	0	80	0	40	0
0	20	0	40	0	80	0	40
40	0	20	0	40	0	80	0
0	40	0	20	0	40	0	80



The mass matrices are same for element 1 and element 2.

⇒ After mapping these two elements mass matrices, we can find the global mass matrix;

$M =$

80	0	40	0	0	0	40	0	20	0	0	0
0	80	0	40	0	0	0	40	0	20	0	0
40	0	160	0	40	0	20	0	80	0	20	0
0	40	0	160	0	40	0	20	0	80	0	20
0	0	40	0	80	0	0	0	20	0	40	0
0	0	0	40	0	80	0	0	0	20	0	40
40	0	20	0	0	0	80	0	40	0	0	0
0	40	0	20	0	0	0	80	0	40	0	0
20	0	80	0	20	0	40	0	160	0	40	0
0	20	0	80	0	20	0	40	0	160	0	40
0	0	20	0	40	0	0	0	40	0	80	0
0	0	0	20	0	40	0	0	0	40	0	80

Initial conditions $\Rightarrow \vec{U}_0 = \vec{U}(0)$, $\dot{\vec{U}}_0 = \dot{\vec{U}}(0)$.

Calculation Steps:

- Equation of motion : $\vec{m}\ddot{\vec{U}} + \vec{K}\vec{U} = \vec{p}(t)$
- Undamped system $\Rightarrow c$ is ignored in all steps.
- $p(0) = 0$, $p(0.1) = -644$ pounds , $p(0.2) = -985$ pounds
 $p(0.3) = -863$ pounds , $p(0.4) = -335$ pounds
- For a fixed beam , system stiffness matrix and mass mat were found previously.

1-) To find $\ddot{\vec{U}}_0 \Rightarrow \vec{m}\ddot{\vec{U}}_0 = \vec{p}_0 - \vec{K}\vec{U}_0$

For $t=0$ force and displacements are zero.

The system has 12 degrees of freedom.

\Rightarrow For the time $= 0 \Rightarrow$ acceleration is zero.

$\Rightarrow \ddot{\vec{U}}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow$ for the time $= 0$;
 every value will be zero.

$$2-) \hat{k} = \vec{k} + \frac{1}{\beta(\Delta t)^2} \vec{m}$$

k and m were found previously. $\beta = 0.25$, $\Delta t = 0.1$
Replacing them into the equation;

⇒ $\hat{k} =$

1.00E+30	178571.4	263252.7	-13736.3	0	0	-588396	13736.26	-349143	-178571	0	0
178571.4	1.00E+30	13736.26	906109.9	0	0	-13736.3	-1802681	-178571	-920571	0	0
263252.7	13736.26	1492571	-2.91E-11	263252.7	-13736.3	-349143	178571.4	-1176791	0	-349143	-178571
-13736.3	906109.9	-2.91E-11	3778286	13736.26	906109.9	178571.4	-920571	0	-3605363	-178571	-920571
0	0	263252.7	13736.26	746285.7	-178571	0	0	-349143	178571.4	-588396	-13736.3
0	0	0	-13736.3	906109.9	-178571	1889143	0	0	178571.4	-920571	13736.26
-588396	-13736.3	-349143	178571.4	0	0	1.00E+30	-178571	263252.7	13736.26	0	0
13736.26	-1802681	178571.4	-920571	0	0	-178571	1.00E+30	-13736.3	906109.9	0	0
-349143	-178571	-1176791	3.64E-12	-349143	178571.4	263252.7	-13736.3	1492571	2.91E-11	263252.7	13736.26
-178571	-920571	-1.82E-12	-3605363	178571.4	-920571	13736.26	906109.9	2.91E-11	3778286	-13736.3	906109.9
0	0	-349143	-178571	-588396	13736.26	0	0	263252.7	-13736.3	746285.7	178571.4
0	0	-178571	-920571	-13736.3	-1802681	0	0	13736.26	906109.9	178571.4	1889143

$$3-) \vec{a} = \frac{1}{\beta \Delta t} \vec{m}, \quad b = \frac{1}{2\beta} \vec{m}$$

Replacing β , Δt and m ;

⇒ $a =$

3.20E+03	0	1600	0	0	0	1600	0	800	0	0	0
0	3.20E+03	0	1600	0	0	0	1600	0	800	0	0
1600	0	6400	0.00E+00	1600	0	800	0	3200	0	800	0
0	1600	0.00E+00	6400	0	1600	0	800	0	3200	0	800
0	0	1600	0	3200	0	0	0	800	0	1600	0
0	0	0	1600	0	3200	0	0	0	800	0	1600
1600	0	800	0	0	0	3.20E+03	0	1600	0	0	0
0	1600	0	800	0	0	0	3.20E+03	0	1600	0	0
800	0	3200	0.00E+00	800	0	1600	0	6400	0.00E+00	1600	0
0	800	0.00E+00	3200	0	800	0	1600	0.00E+00	6400	0	1600
0	0	800	0	1600	0	0	0	1600	0	3200	0
0	0	0	800	0	1600	0	0	0	1600	0	3200

⇒ b =

1.60E+02	0	80	0	0	0	80	0	40	0	0	0
0	1.60E+02	0	80	0	0	0	80	0	40	0	0
80	0	320	0.00E+00	80	0	40	0	160	0	40	0
0	80	0.00E+00	320	0	80	0	40	0	160	0	40
0	0	80	0	160	0	0	0	40	0	80	0
0	0	0	80	0	160	0	0	0	40	0	80
80	0	40	0	0	0	1.60E+02	0	80	0	0	0
0	80	0	40	0	0	0	1.60E+02	0	80	0	0
40	0	160	0.00E+00	40	0	80	0	320	0.00E+00	80	0
0	40	0.00E+00	160	0	40	0	80	0.00E+00	320	0	80
0	0	40	0	80	0	0	0	80	0	160	0
0	0	0	40	0	80	0	0	0	80	0	160

- The first three step should be done one time for an analysis. After step three, it should be calculated for each time step.

⇒ For each time step :

$$4-) \hat{\Delta \vec{p}} = \Delta \vec{p}_i + \vec{a} \vec{U}_i + \vec{b} \vec{U}_i$$

At the time step 0, there are no displacement, velocity, acceleration due to $p(0.0) = 0$.

$$i=1 \Rightarrow t=0.1s \Rightarrow p(0.1) = -644 \text{ pounds.}$$

The dynamic load is acting at the 6th node vertically and we have 12 d.o.f.s.

$$\Rightarrow \text{force matrix at } 0.1s \Rightarrow p(0.1) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -644 \end{pmatrix}$$

$$\Rightarrow \text{at } 0.1s \Rightarrow i=1 \quad \vec{p}_0 = 0 \Rightarrow \Delta \vec{p}_1 = \vec{p}_1 = p(0.1)$$

initially \vec{u} and $\dot{\vec{u}}$ are zero.

$$\Rightarrow \text{the equation yields} \Rightarrow \Delta \vec{p}^1 = \Delta \vec{p}_1 = p(0.1)$$

$$\Rightarrow \Delta \vec{p}^1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -644 \end{pmatrix}$$

$$5-) \quad \hat{\vec{k}} \Delta \vec{U}_i = \Delta \hat{\vec{p}}_i$$

$\hat{\vec{k}}$ and $\Delta \hat{\vec{p}}_i$ are known, by using gauss elimination;

$$\Delta \vec{U}_i = \begin{pmatrix} 0 \\ 0 \\ -2.84 \times 10^{-4} \\ 3.73 \times 10^{-4} \\ -7.46 \times 10^{-4} \\ -5.9 \times 10^{-3} \\ 0 \\ 0 \\ 2.1 \times 10^{-4} \\ 4.2 \times 10^{-4} \\ 8.7 \times 10^{-4} \\ -6.1 \times 10^{-3} \end{pmatrix} \text{ (inch)}$$

$$6-) \quad \Delta \vec{U}_i = \frac{\kappa}{\beta \Delta t} \Delta \vec{U}_i - \frac{\kappa}{\beta} \vec{U}_i + \Delta t \left(1 - \frac{\kappa}{2\beta}\right) \ddot{\vec{U}}_i$$

Initially $\ddot{\vec{U}}$ and \vec{U} are zero.

$$\Rightarrow \text{equation yields} \rightarrow \Delta \vec{U}_i = \frac{0.5}{(0.25)(0.1)} \Delta \vec{U}_i$$

was found previous step.

$$\Rightarrow \Delta \vec{U}_i = \begin{pmatrix} 0 \\ 0 \\ -0.0057 \\ 0.0075 \\ -0.0149 \\ -0.1181 \\ 0 \\ 0 \\ 0.0042 \\ 0.0084 \\ 0.0174 \\ -0.1222 \end{pmatrix}$$

$$7-) \Delta \vec{U}_i = \frac{1}{\beta (\Delta t)^2} \Delta \vec{U}_i - \frac{1}{\beta \Delta t} \vec{U}_i - \frac{1}{2\beta} \vec{U}_i$$

Initially \vec{U} and \vec{U} are zero.

$$\Rightarrow \text{equation yields } \Delta \vec{U}_1 = \frac{1}{(0.25)(0.1)^2} \Delta \vec{U}_1$$

↓
was found in step 5

$$\Rightarrow \Delta \vec{U}_1 = \begin{pmatrix} 0 \\ 0 \\ -0.1137 \\ 0.1495 \\ -0.2987 \\ -2.3617 \\ 0 \\ 0 \\ 0.0840 \\ 0.1689 \\ 0.3490 \\ -2.4447 \end{pmatrix}$$

8-) In this step; the calculated change of displacements, velocities and accelerations will be added to previous displacements and accelerations.

$$\left. \begin{aligned} \vec{U}_{i+1} &= \vec{U}_i + \Delta \vec{U}_i \\ \vec{U}_{i+1} &= \vec{U}_i + \Delta \vec{U}_i \\ \vec{U}_{i+1} &= \vec{U}_i + \Delta \vec{U}_i \end{aligned} \right\} \Rightarrow \text{Initially } \vec{U}, \vec{U} \text{ and } \vec{U} \text{ are zeros. So, for the } t=0.1$$

$$\left. \begin{aligned} \vec{U}_1 &= \Delta \vec{U}_1 \\ \vec{U}_1 &= \Delta \vec{U}_1 \\ \vec{U}_1 &= \Delta \vec{U}_1 \end{aligned} \right\} \Rightarrow \text{they found at steps 5, 6 and 7.}$$

To complete the analysis, the steps 4-8 should be repeated for each time steps.

- For $t = 0.2 \text{ s} \Rightarrow p(0.2) = -985 \text{ pounds}$.

Repeating the steps 4 to 8 for $p(0.2)$;

$$\vec{p}_1 = -644 \text{ pounds} \quad \text{and} \quad \vec{p}_2 = -985 \text{ pounds}$$

$$\Rightarrow \Delta \vec{p} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -341 \end{pmatrix} \quad \text{and} \quad \Delta \hat{p} = \Delta \vec{p} + \underbrace{\vec{a} \vec{u}_1 + \vec{b} \vec{u}_1}_{\substack{\downarrow \\ \text{found previously.}}}$$

$$\Rightarrow \Delta \hat{p} = \begin{pmatrix} -11.46 \\ 37.42 \\ -65.73 \\ -423.74 \\ -51.22 \\ -1109.5 \\ 4.35 \\ 38.97 \\ 49.33 \\ -424.17 \\ 68.21 \\ -1462.2 \end{pmatrix}$$

$$\frac{\Delta}{k} \Delta \vec{U}_i = \Delta \vec{P}_i \quad \text{by using gauss elimination;}$$

$$\Delta \vec{U}_2 = \begin{Bmatrix} 0 \\ 0 \\ -0.0011 \\ -0.0014 \\ -0.0027 \\ -0.0236 \\ 0 \\ 0 \\ 0.0011 \\ -0.0014 \\ 0.0028 \\ -0.0237 \end{Bmatrix} \quad (\text{inch})$$

$$\Delta \vec{U}_2 = \frac{\gamma}{\beta \Delta t} \Delta \vec{U}_2 - \frac{\gamma}{\beta} \vec{U}_1 + \Delta t \left(1 - \frac{\gamma}{2\beta}\right) \vec{U}_i$$

\downarrow
 found above

\swarrow
 were found previous se

$$\Rightarrow \Delta \vec{U}_2 = \begin{Bmatrix} 0 \\ 0 \\ -0.011 \\ -0.042 \\ -0.023 \\ -0.023 \\ 0 \\ 0 \\ 0.012 \\ -0.044 \\ 0.021 \\ -0.23 \end{Bmatrix}$$

$$\Delta \vec{U}_2 = \frac{1}{\beta(\Delta t)^2} \Delta \vec{U}_2 - \frac{1}{\beta \Delta t} \vec{U}_1 - \frac{1}{2\beta} \vec{U}_1$$

was found previous page

were found at the previous step.

$$\Rightarrow \Delta \vec{U}_2 = \begin{Bmatrix} 0 \\ 0 \\ 0.003 \\ -1.14 \\ 0.12 \\ 0.005 \\ 0 \\ 0 \\ 0.087 \\ -1.21 \\ -0.26 \\ 0.29 \end{Bmatrix}$$

$$\vec{U}_2 = \vec{U}_1 + \Delta \vec{U}_2, \quad \vec{\dot{U}}_2 = \vec{\dot{U}}_1 + \Delta \vec{\dot{U}}_2, \quad \vec{\ddot{U}}_2 = \vec{\ddot{U}}_1 + \Delta \vec{\ddot{U}}_2$$

$$\Rightarrow \vec{U}_2 = \begin{Bmatrix} 0 \\ 0 \\ -0.001 \\ -0.001 \\ -0.003 \\ -0.029 \\ 0 \\ 0 \\ 0.001 \\ -0.0009 \\ 0.003 \\ -0.0298 \end{Bmatrix}, \quad \vec{\dot{U}}_2 = \begin{Bmatrix} 0 \\ 0 \\ -0.016 \\ -0.035 \\ -0.038 \\ -0.354 \\ 0 \\ 0 \\ 0.017 \\ -0.0355 \\ 0.0393 \\ -0.352 \end{Bmatrix}, \quad \vec{\ddot{U}}_2 = \begin{Bmatrix} 0 \\ 0 \\ -0.11 \\ -1.00 \\ -0.172 \\ -2.355 \\ 0 \\ 0 \\ 0.171 \\ -1.048 \\ 0.088 \\ -2.154 \end{Bmatrix}$$

- For $t = 0.3 \text{ s} \implies p(0.3) = -863 \text{ pounds.}$

Again repeating the steps 4 to 8 for $p(0.3)$;

$$\vec{p}_2 = -985 \text{ pounds} \quad \text{and} \quad \vec{p}_3 = -863 \text{ pounds}$$

$$\Rightarrow \Delta \vec{P} = \begin{Bmatrix} 0 \\ 122 \end{Bmatrix} (\text{pounds}) \quad \Rightarrow \hat{\Delta P} = \begin{Bmatrix} -15.36 \\ -206.45 \\ -101.8 \\ -1948.5 \\ -96.06 \\ -2451.9 \\ 27.03 \\ -208.73 \\ 124.48 \\ -1948.2 \\ 87.72 \\ -2313.3 \end{Bmatrix} (\text{pounds})$$

(from step 4)

from step 5: $\Delta \vec{U}_3 = \begin{Bmatrix} 0 \\ 0 \\ -0.002 \\ -0.01 \\ -0.003 \\ -0.042 \\ 0 \\ 0 \\ 0.002 \\ -0.01 \\ 0.003 \\ -0.042 \end{Bmatrix}$, and from step 6: $\Delta \vec{U}_3 = \begin{Bmatrix} 0 \\ 0 \\ -0.01 \\ -0.146 \\ 0.005 \\ -0.139 \\ 0 \\ 0 \\ 0.012 \\ -0.146 \\ -0.007 \\ -0.142 \end{Bmatrix}$

from step 7 :

$$\Delta \vec{U}_3 = \begin{pmatrix} 0 \\ 0 \\ 0.002 \\ -0.932 \\ 0.448 \\ 1.914 \\ 0 \\ 0 \\ -0.096 \\ -0.823 \\ -0.333 \\ 1.46 \end{pmatrix}$$

From step 8 :

$$\vec{U}_3 = \begin{pmatrix} 0 \\ 0 \\ -0.003 \\ -0.011 \\ -0.007 \\ -0.071 \\ 0 \\ 0 \\ 0.003 \\ -0.011 \\ 0.007 \\ -0.072 \end{pmatrix}, \quad \vec{U}_3 = \begin{pmatrix} 0 \\ 0 \\ -0.027 \\ -0.181 \\ -0.033 \\ -0.493 \\ 0 \\ 0 \\ 0.029 \\ -0.181 \\ 0.031 \\ -0.494 \end{pmatrix}, \quad \vec{U}_3 = \begin{pmatrix} 0 \\ 0 \\ -0.1 \\ -1.933 \\ 0.276 \\ -0.441 \\ 0 \\ 0 \\ 0.075 \\ -1.871 \\ -0.244 \\ -0.694 \end{pmatrix}$$

- For $t = 0.4 \text{ s} \Rightarrow p(0.4) = -335$

Again repeating the steps 4 to 8 for $p(0.4)$;

$\vec{p}_3 = -863$ pounds and $\vec{p}_4 = -335$ pounds

$$\Rightarrow \Delta \vec{p} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 528 \end{pmatrix} \Rightarrow \Delta \vec{p}^{\wedge} = \begin{pmatrix} -26.6 \\ -665.4 \\ -122.1 \\ -3910.6 \\ -58.1 \\ -3163.3 \\ 26.5 \\ -662.8 \\ 121.2 \\ -3910.7 \\ 57 \\ -2654.1 \end{pmatrix} \quad (\text{step 4})$$

Step 5:

$$\Delta \vec{U}_4 = \begin{pmatrix} 0 \\ 0 \\ -0.003 \\ -0.024 \\ -0.002 \\ -0.049 \\ 0 \\ 0 \\ 0.003 \\ -0.024 \\ 0.002 \\ -0.049 \end{pmatrix}$$

Step 6:

$$\Delta \vec{U}_4 = \begin{pmatrix} 0 \\ 0 \\ -0.005 \\ -0.12 \\ 0.019 \\ -0.007 \\ 0 \\ 0 \\ 0.003 \\ -0.121 \\ -0.018 \\ -0.002 \end{pmatrix}$$

Step 7 :

$$\Delta \vec{U}_4 = \begin{pmatrix} 0 \\ 0 \\ 0.098 \\ 1.452 \\ -0.157 \\ 0.733 \\ 0 \\ 0 \\ -0.075 \\ 1.303 \\ 0.122 \\ 1.335 \end{pmatrix}$$

Step 8 :

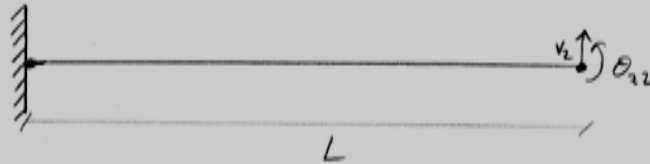
$$\vec{U}_4 = \begin{pmatrix} 0 \\ 0 \\ -0.006 \\ -0.036 \\ -0.009 \\ -0.121 \\ 0 \\ 0 \\ 0.006 \\ -0.036 \\ 0.009 \\ -0.121 \end{pmatrix}, \quad \vec{U}_4 = \begin{pmatrix} 0 \\ 0 \\ -0.033 \\ -0.302 \\ -0.013 \\ -0.501 \\ 0 \\ 0 \\ 0.033 \\ -0.303 \\ 0.013 \\ -0.497 \end{pmatrix}, \quad \vec{U}_4 = \begin{pmatrix} 0 \\ 0 \\ -0.009 \\ -0.48 \\ 0.118 \\ 0.291 \\ 0 \\ 0 \\ -0.0002 \\ -0.567 \\ -0.122 \\ 0.641 \end{pmatrix}$$

Time Step :	t = 0.1 second		t = 0.2 second	
DOF	Hand Calculation Results	Computer Results	Hand Calculation Results	Computer Results
u1	0	2.29E-28	0	6.24E-28
u2	0	9.15E-29	0	3.37E-28
u3	-0.00028	-0.00028	-0.001	-0.00141
u4	0.00037	0.000374	-0.001	-0.00101
u5	0.00074	-0.00075	-0.003	-0.00342
u6	-0.0059	-0.0059	-0.029	-0.02951
u7	0	-2.27E-28	0	-6.31E-28
u8	0	1.51E-29	0	2.27E-28
u9	0.00021	0.00021	0.001	0.00127
u10	0.00042	0.000422	-0.0009	-0.00093
u11	0.00087	0.000872	0.003	0.003711
u12	-0.0061	-0.00611	-0.0298	-0.02983

Time Step:	t = 0.3 second		t = 0.4 second	
DOF	Hand Calculation Results	Computer Results	Hand Calculation Results	Computer Results
u1	0	-8.05E-29	0	-2.87E-27
u2	0	4.09E-28	0	-4.15E-29
u3	-0.003	-0.00365	-0.006	-0.00672
u4	-0.011	-0.01184	-0.036	-0.03605
u5	-0.007	-0.00701	-0.009	-0.00935
u6	-0.071	-0.0719	-0.121	-0.12166
u7	0	8.65E-29	0	2.87E-27
u8	0	3.02E-28	0	-8.43E-29
u9	0.003	0.003589	0.006	0.006715
u10	-0.011	-0.01178	-0.036	-0.03603
u11	0.007	0.007254	0.009	0.009488
u12	-0.072	-0.07217	-0.121	-0.12177

Comparison of Analytical and Computer Results:

For Case 1 in Chapter 4:



$$E = 4.35 \times 10^6 \text{ psi} \quad , \quad \rho = 0.0868 \text{ lb/in}^3$$

$$L = 144 \text{ in} \quad , \quad \text{depth} = 12 \text{ in} \quad , \quad \text{thickness} = \text{unity (1 in)}$$

$$\Rightarrow I = 144 \text{ in}^4 \quad , \quad A = 1.0416 \text{ in}^2 \text{ (cross section area)}$$

$$\text{Loaded at the free end ; } F = 1800 \sin(0.15t)$$

$$\text{Time duration} = 10 \text{ seconds} \quad , \quad \Delta t = 0.1 \text{ second}$$

Because of fixity at the left side, we can neglect v_1 and θ_{21} .

\Rightarrow Stiffness and mass matrices can be written 2×2 matrices.

$$m\ddot{u} + ku = p(t)$$

$$k = \frac{EI}{L^3} \begin{bmatrix} 12 & -6L \\ -6L & 4L^2 \end{bmatrix} \text{ and } m = \overset{\text{total mass}}{\frac{m}{420}} \begin{bmatrix} 156 & -22L \\ -22L & 4L^2 \end{bmatrix}$$

$$\text{total mass} = \rho AL = (0.0868)(1.0416)(144) = 150 \text{ lbs}$$

$$\Rightarrow k = \begin{bmatrix} 2517.4 & -181250 \\ -181250 & 17400000 \end{bmatrix}$$

$$\text{and } m = \begin{bmatrix} 55.7143 & -1131.4 \\ -1131.4 & 29623.0 \end{bmatrix}$$

Harmonic force can be written;

$$\begin{Bmatrix} p_0 \\ 0 \end{Bmatrix} \sin \omega t = \begin{Bmatrix} 1800 \\ 0 \end{Bmatrix} \sin(0.15t)$$

$$\Rightarrow \omega = 0.15 \quad \text{and} \quad p_0 = 1800$$

$$\begin{Bmatrix} v_2(t) \\ \theta_{22}(t) \end{Bmatrix} = \begin{Bmatrix} v_{20} \\ \theta_{20} \end{Bmatrix} \sin \omega t$$

The equation of motion becomes ;

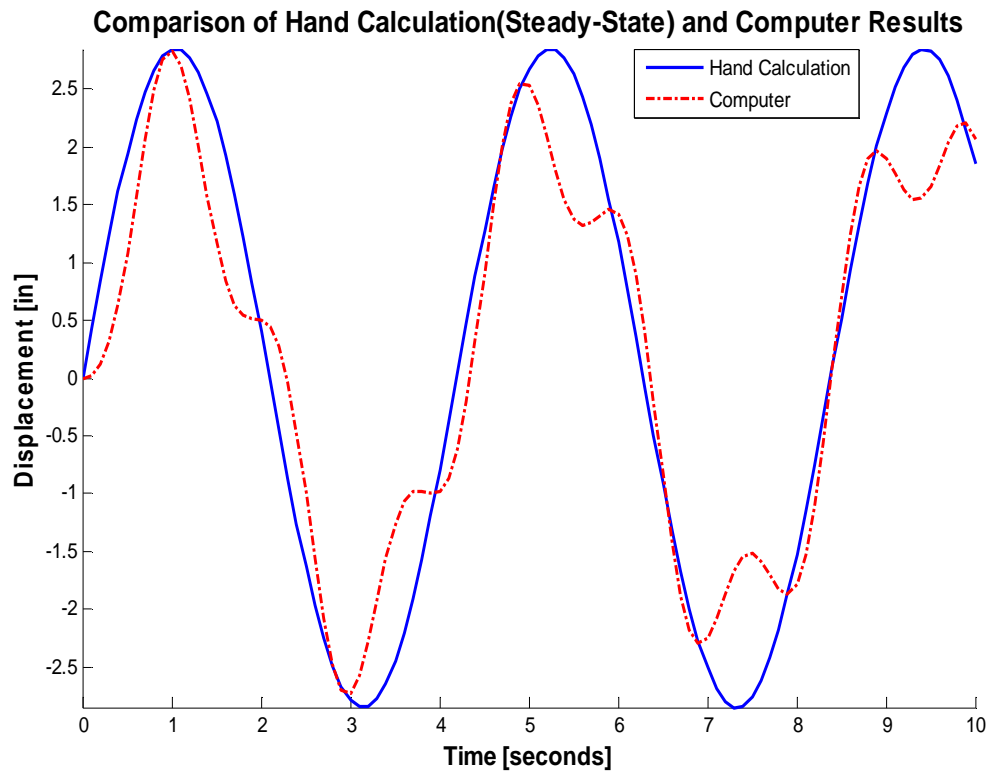
$$[k - \omega^2 m] \begin{Bmatrix} v_{10} \\ \theta_{220} \end{Bmatrix} = \begin{Bmatrix} p_0 \\ 0 \end{Bmatrix}$$

Substituting the previously found values into the this equation and solving it ;

$$\Rightarrow \boxed{v_{10} = 2.8508 \sin(0.15t)} \text{ (in)}$$

This gives vertical displacement function at the free end with respect to time. (Steady-State response)

The illustration of comparison is given as a figure
For the computer results, node 39 - d.o.f. 78
is used from chapter 4 - case 1.



Note: The displacement response shown compares the steady-state solution of the hand-calculation only. The computer program results include both the steady-state response and the transient response of the system. As a result, some disagreement, particularly at the peaks of the steady-state response, is expected and confirmed in the figure.

VITA

Muhammet Ali Saglar

Candidate for the Degree of

Master of Science

Thesis: DEVELOPMENT OF A FINITE ELEMENT ANALYSIS PROGRAM FOR
STRUCTURAL DYNAMICS APPLICATIONS

Major Field: Civil Engineering

Biographical:

Personal Data:

Born in Balikesir, Turkey, August 22nd 1984, son of Remzi Saglar and Asiye Saglar.

Education:

Graduated high school from Ataturk Anatolian, Ankara, Turkey in June 2002; received Bachelor of Engineering degree in Civil Engineering from Istanbul Technical University, Istanbul, Turkey in June 2007. Completed the requirements for the Master of Science in Civil Engineering at Oklahoma State University, Stillwater, Oklahoma in July, 2009.

Experience:

Employed by Oklahoma State University as a research assistant from January 2008 to July 2009. Employed by Tekfen Corporation as a civil engineering intern from May 2006 to July 2006. Employed by Yuksel Construction Corporation as a civil engineering intern from May 2005 to July 2005.

Professional Memberships:

American Concrete Institute, American Institute of Steel Construction.

Name: Muhammet Ali Saglar

Date of Degree: July, 2009

Institution: Oklahoma State University

Location: Stillwater, Oklahoma

Title of Study: DEVELOPMENT OF A FINITE ELEMENT ANALYSIS PROGRAM
FOR STRUCTURAL DYNAMICS APPLICATIONS

Pages in Study: 107

Candidate for the Degree of Master of Science

Major Field: Civil Engineering

Scope and Method of Study:

The finite element method is a powerful method to find solutions for engineering problems. Structural dynamics is an important concept for understanding behavior of structures under different types of dynamic loads. The combination of these two concepts into a seamless, integrated computer program to analyze structural systems considering dynamic, time-dependent loads provides tremendous capabilities with respect to numerical analysis. The implementation of the theoretical development of both concepts into a fully-functional MATLAB computer program provided the primary objective of this study.

Findings and Conclusions:

The development of the MATLAB computer program to analyze a structural system using the finite element method incorporating structural dynamic response due to time-varying loads was accomplished. Several case studies were presented that illustrate the capabilities of this program with respect to determining the response of the structural system. In the current study, bilinear quadrilateral (Q4) isoparametric finite elements were used to construct the finite element mesh of the structural system. The finite element method was then integrated with a numerical time-stepping method, the Newmark-Beta method, to determine the response of the structural system due to a dynamic loading.

ADVISER'S APPROVAL: Dr. Jonathan S. Goode